THE UNIVERSITY OF PARDUBICE

FACULTY OF TRANSPORT ENGINEERING

DOCTORAL THESIS

2025 JAN BERG

THE UNIVERSITY OF PARDUBICE

Faculty of Transport Engineering

Department of Transport Means and Diagnostics

Development of a Methodology of a Traffic Monitoring Using Video Recordings for Traffic Accident Analysis

Doctoral Thesis

Ing. Jan Berg

2025

Student

Ing. Jan Berg

Study Programme

Transport Means and Infrastructure

Field of Study

Transport Means

Supervisor

prof. Ing. Jan Krmela, Ph.D.

Co-supervisor

Ing. Jan Pokorný, Ph.D.

Department

Department of Transport Means and Diagnostics

I hereby declare:

The thesis entitled „*Development of a Methodology of a Traffic Monitoring Using Video Recordings for Traffic Accident*“ is my own work. All literal sources and information that I used in the thesis are referenced in the bibliography.

I have been acquainted with the fact that my work is subject to the rights and obligations arising from Act No. 121/2000 Sb., On Copyright, on Rights Related to Copyright and on Amendments to Certain Acts (Copyright Act), as amended, especially with the fact that the University of Pardubice has the right to conclude a license agreement for the use of this thesis as a school work under Section 60, Subsection 1 of the Copyright Act, and that if this thesis is used by me or a license to use it is granted to another entity, the University of Pardubice is entitled to request a reasonable fee from me to cover the costs incurred for the creation of the work, depending on the circumstances up to their actual amount.

I acknowledge that in accordance with Section 47b of Act No. 111/1998 Sb., On Higher Education Institutions and on Amendments to Other Acts (Higher Education Act), as amended, and the Directive of the University of Pardubice No. 7/2019 Rules for Submission, Publication and Layout of Theses, as amended, the thesis will be published through the Digital Library of the University of Pardubice.

This thesis was accomplished with the support of technologies of the Educational and Research Centre in Transport.

In Pardubice on 5th September 2025

……
Jan Berg

ACKNOWLEDGEMENTS:

The following several rows are assigned to all who allowed me to study and successfully finish the doctoral program. First of all, I would like to thank a lot to the supervisor of my thesis, prof. Ing. Jan Krmela, Ph.D., for his great guidance and trust. Big thanks also belong to my co-supervisor Ing. Jan Pokorný, Ph.D. for his support with physical measurements and consultations in terms of the expert praxis of the traffic accidents analysis together with Ing. Zdeněk Mrázek, Ph.D.

I would like to express a great gratitude to my girlfriend Ing. Kristýna Exnerová for her continuous encouragement during the studies and providing motivation and willingness to continue. I would also like to thank a lot to my family for their great support during all my studies. Very appreciated was a help and great cooperation on publications with my colleagues from the university, namely Ing. Petr Jilek, dis. Ph.D. and Ing. Sadjiep Tchuigwa Baurice Sylvain, Ph.D. I thank a lot to Ing. Jakub Vágner, Ph.D. for providing me all the needed HW and computational equipment. Big thanks belong to the Faculty of Transportation Sciences CTU in Prague, Department of Forensic Experts in Transportation, for providing the GNSS station that enabled to validate the tracker SW. I acknowledge with gratitude the support of Ing. Petr Kvaš and Lenka Uskobová from the City Police Department in Pardubice, who provided me with the video recordings from the crossings.

ANOTATION

The doctoral thesis „*Development of a Methodology of a Traffic Monitoring Using Video Recordings for Traffic Accident*“ introduces a complex method for automatic computational evaluation of a selected dynamic quantity used in a traffic accidents analysis praxis from video recording from traffic surveillance cameras. The procedure follows the selection of suitable input data in form of video recording and a robust detection and tracking SW implementation and fine-tuning for the particular use. A calibration method is developed and tailored to the use case in the thesis. After obtaining the data, a suitable data processing and smoothening methods are introduced and implemented. As a conclusion, a statistical processing of a non-sudden braking is provided which has a real use in the traffic accidents analysis praxis.

KEYWORDS

automatic traffic analysis, automatic image processing, non-sudden braking

ANOTACE

Disertační práce „Vypracování metodiky sledování provozu pomocí kamerového záznamu k využití pro analýzu dopravních nehod“ představuje komplexní metodu pro automatické výpočetní vyhodnocení vybrané dynamické veličiny využívané v praxi analýzy dopravních nehod z videozáznamů z dopravních kamer. Postup zahrnuje výběr vhodných vstupních dat ve formě videozáznamu a implementaci robustního softwaru pro detekci a sledování objektů, včetně jeho optimalizace pro konkrétní použití. V práci je dále vyvinuta a na daný případ přizpůsobena metoda kalibrace obrazu. Po získání dat jsou představeny a implementovány vhodné metody jejich zpracování a vyhlazování. V závěru je provedeno statistické zpracování veličiny nenáhlého brzdění, které má reálné uplatnění v praxi analýzy dopravních nehod.

KLÍČOVÁ SLOVA

automatická analýza dopravy, automatické zpracování obrazu, nenáhlé brzdění

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF GRAPHS

LIST OF ABBREVIATIONS

Abbreviation	Meaning
AI	Artificial Intelligence
AP	Average Precision
BB	Boounding Box
BoT-SORT	Bag of Tricks-Simple Online and Realtime Tracking
CLIP	Contrastive Language-Image Pretraining
CNNs	Convolutional Neural Networks
COCO	Common Objects in Context
CTU	Czech Technical University (ČVUT)
ČÚZK	Český úřad zeměměřický a katastrální
DeepSORT	Deep Simple Online and Realtime Tracking
DEFT	Detection Embeddings for Tracking
DETR	DEtection TRansformer
DINO	DETR with Improved DeNoising and anchor boxes
DL	Deep learning
DLT	Direct Linear Transformation
EDA	Estimation of Distribution Algorithm
E-ELAN	Extended Efficient Layer Aggregation Network
FMEA	Failure Mode and Effects Analysis
FN	False Negative
FP	False Positive
GMM	Gaussian Mixture Model
GMMCM	Gaussian Mixture Modelling with Confidence Measurement
GNSS	Global Navigation Satellite System station
GOTURN	Generic Object Tracking Using Regression Networks
GPUs	Graphics Processing Units
IoU	Intersection of Union
KLT tracker	Kanade–Lucas–Tomasi tracker
LiDAR	Light Detection and Ranging
LSTM	Long Short Term Memory
mAP	mean Average Precision
MDNet	Multi-Domain Network
MOG2	Mixture of Gaussians, version 2
MOT	Multiple object tracking

MOTA	Multiple Object Tracking Accuracy
MOTP	Multiple Object Tracking Precision
MS-COCO	Microsoft Common Objects in Context
NMS	Non-Maximum Suppression
NRMSE	Normalized Root Mean Square Error
OC-SORT	Observation-Centric Simple Online and Realtime Tracking
OpenCV	Open Source Computer Vision Library
OWL-ViT	Open-World Localization with Vision Transformers
PHOG	Pyramid Histogram of Oriented Gradients
PnP	Perspective-n-Point
PyCharm IDE	PyCharm Integrated Development Environment
Q-Q	Quantile-Quantile
RANSAC	RANdom SAmple Consensus
R-CNN	Region-based CNN
RepConvN	Re-parameterized Convolution Network
R-FCN	Region-based Fully Convolutional Network
RGB	Red-Green-Blue
RMSE	Root Mean Square Error
RoIs	Regions of Interest
ROLO	Recurrent YOLO
SAM	Segment Anything Model
S-G	Savitzky-Golay
SORT	Simple Online and Realtime Tracking
SOT	Single Object Tracking
SSD	Single Shot Detector
SVM	Support Vector Machine
SW	software
TP	True Positive
ViBe	Visual Background Extractor
ViT	Vision Transformer
VP	Vanishing Point acquisition
YOLO	You Only Look Once

1. INTRODUCTION

When solving road accidents, a knowledge of kinematic and dynamic quantities, that represent a common driver behaviour in non-sudden situations, is often required. These quantities are then compared to those evaluated from the situation to determine, how sudden situation it was for drivers. Nevertheless, each driver has different driving skills, experience, pace, in which they drive, some drives aggressively and some conversely defensively etc. Their driving behaviour depends also on their age, gender, tiredness, and other internal and external factors. Moreover, in the forensic praxis of traffic accidents analysis, there is no clear definition where the value of the threshold between sudden and non-sudden reaction lies. There are multiple different assumptions introduced and applied.

The aim of the thesis was to obtain data describing a normal behaviour of drivers that are fully independent and, what is important, without awareness of the drivers, whose drives were measured. In order to obtain such a data sample effectively, an own software is developed. This software is able to go through surveillance videos of desired traffic locations automatically and evaluate desired kinematic and dynamic quantities. The accuracy of the software is verified by experiments, when quantities computed by the software and the same quantities measured physically in the vehicle are compared.

This doctoral thesis also includes a review of state-of-the-art methods for automatic detection and tracking of objects, calibration, data processing etc. In order to fulfil goals of the thesis, the most suitable methods are selected, adjusted to the particular case and merged in a complex procedure.

2. OBJECTIVES OF THE DISSERTATION THESIS

In order to achieve the purpose of this thesis, several specific objectives were set and are presented in this chapter.

- **Develop a method suitable for automatic traffic analysis.**
- **Provide a particular application of the method.**
- **Validate a precision of used method.**
- **Evaluate a non-sudden braking deceleration quantity for use in the traffic accidents analysis praxis.**

Formulation of a scientific hypothesis: The developed system based on an automated traffic analysis is capable of extracting kinematic and dynamic parameters of vehicles from traffic camera recordings with sufficient precision to be applicable for analysis of driver behaviour in real traffic conditions. (The hypothesis will be either confirmed or rejected in the end of the thesis).

3. Review of Theoretical Background and Related Work

With the growth of availability of computational power, increasingly more processes, which have been previously performed only by hand, can be automated and thus more data can be processed in a short time at a sufficient accuracy. Nowadays problems are solved not only in the field of automatic traffic analysis from surveillance cameras, but also from on-board cameras from vehicles. Such detection must work in real-time which is challenging considering the fact that the camera is actively moving with the vehicle. In this thesis only selected methods concerning static camera records analysis will be provided, because traffic cameras are mostly firmly mounted and focused on particular regions of interest.

As mentioned, authors currently use various methods to detect and track vehicles. The mostly used methods that have high accuracy are methods based on background subtracting and methods based on deep learning and neural networks. In this thesis mainstream methods of the research will be described.

3.1. Automatic road user detection

In order to be able to automatically analyse traffic in the desired area, it is necessary that the computer „sees“ all the relevant objects. There are a few terms concerning the detection that have to be clarified.

Detection and **recognition** mean that there is somewhere an object in the image.

Localization stands for a process of finding its coordinates within the image.

Classification means the affiliation of the object to some category.

In the following text the term detection will for simplicity include all the processes of detection (recognition), classification and localization. The basic mainstream approaches will be presented and described in the text below.

Detection accuracy metrics

The detection (identification or as well recognition) of the object is measured by IoU (Intersection over Union), which calculation is represented in the Figure 1.

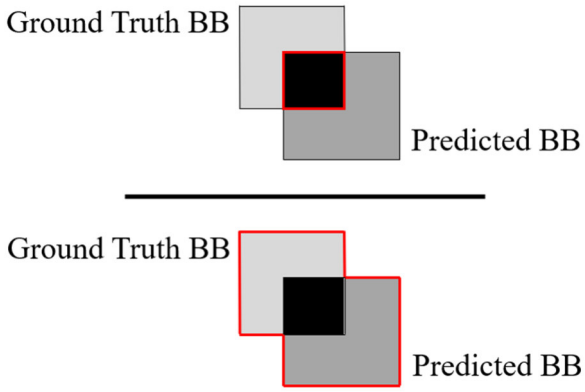


Figure 1 – Intersection over Union (IoU) (author)

In the numerator there is an intersection of areas of the Ground Truth bounding box and predicted bounding box and the denominator contains the whole area of both bounding boxes overlapping each other. The value of IoU, that always lies in the interval from 0 to 1, can be classified to several categories:

- IoU ≥ 0.5 — True Positive (TP).
- IoU < 0.5 — False Positive (FP).
- No detection of present object — False Negative (FN).

By the True Positives the object detection is considered to be successful, as the bounding boxes overlap well. The threshold of 0.5 can be adjusted to any desired value, while common values when evaluating the IoUs are 0.50-0.95.

Further, there are two quantities calculating with IoU that evaluate the detection accuracy. The formulas 1, 2 containing these quantities are stated below.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{1}$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{2}$$

Finally, from these two quantities the final one is calculated representing the area under the Precision-Recall curve. This way an Average Precision (AP) is calculated for each class of the object. To get the overall characteristic over all classes, a standard mean of all Average Precision is calculated, which is labelled as mAP. (Khandelwal, 2020; Koech, 2020)

3.1.1. Background subtraction

This method is one of the simplest methods for detecting objects in the video records. It is based on the principle of separating the static background from the moving foreground. Its first use and implementation to computer vision can be dated to 1980s and 1990s e.g., Schirra et al. (1987): *From image sequences to natural language: a First step toward automatic perception and description of motions*; Karmann (1990): *Moving object recognition using an adaptive background memory*; Huang et al. (1993): *Symbolic Traffic Scene Analysis Using Dynamic Belief Networks*; Koller et al. (1993): *Robust Multiple Car Tracking with Occlusion Reasoning*; Ridder et al. (1995): *Adaptive Background Estimation and Foreground Detection using Kalman-Filtering* etc.

Models using a standard GMM (Gaussian Mixture Modelling)

As this method has been developed further, new pre-processing methods were added prior to the main detection method in order to improve the detection results. Also, the main detection method of background subtraction changed. After ca. 10 years a method of „Mixture of Gaussians“ emerged and is based on two papers, first Zivkovic (2004) named „*Improved Adaptive Gaussian Mixture Model for Background Subtraction*“ and the other Zivkovic and Heijden (2006) named „*Efficient adaptive density estimation per image pixel for the task of background subtraction*“. Gaussian Mixture Modelling (GMM) clusters the data, where the intensity of each pixel is taken as an input. Then, each point gets associated with one cluster.

Several enhancements have been proposed to improve GMM's robustness to illumination changes, camera noise, and background dynamics. For instance, Mukherjee and Das (2013) introduced an adaptive GMM approach that modifies learning rates and includes a shadow detection step based on the Horpresert color model. This method adapts well to changing lighting conditions and moving background elements. Zhang et al. (2016) presented a variation known as GMM with Confidence Measurement (GMMCM) that evaluates the confidence in each pixel's classification to reduce the influence of slowly moving or temporarily stopped vehicles. Their experiments showed improved accuracy in urban traffic scenes, addressing a common weakness of basic GMM. Aris et al. (2025) proposed an improved GMM with Cuckoo Search Optimization that adapts GMM parameters dynamically for different traffic densities. Their system incorporates exponential decay and outlier filtering to stabilize detection in both low and high congestion scenarios.

Alternative Background Modelling Approaches

While GMM is dominant, other innovative background modelling strategies have emerged. The ViBe algorithm, introduced by Barnich and Van Droogenbroeck (2009), relies on stochastic pixel sampling and non-deterministic updates. Though not GMM-based, it achieves competitive results in noisy or dynamic scenes by storing a history of pixel values and randomly updating them. Magee (2004) employed a per-pixel GMM model combined with motion estimation for tracking multiple vehicles in traffic footage. This method emphasizes not only static background modelling but also the integration of motion information to enhance detection reliability.

Feature-Enhanced and Multi-Modal Techniques

Recent trends also explore combining background subtraction with other sensory modalities or feature sets. For example, Song et al. (2014) fused RGB color data with depth information from stereo or structured light sensors in a GMM framework. This dual-modality approach significantly reduced errors in camouflaged or low-contrast environments. In pedestrian detection, Alom and Taha (2017) integrated adaptive GMM with a neural network classifier (based on PHOG features) and Kalman filters for multi-view pedestrian tracking. This hybrid system showcases how traditional background subtraction can serve as a robust base layer for more advanced object recognition systems.

3.1.2. Deep learning methods

Methods based on DL (deep learning) have been significantly developed since 2006 with the emerge of high performance parallel computational systems (GPUs). The main principle compared to traditional programming is shown in the Figure 2 below:

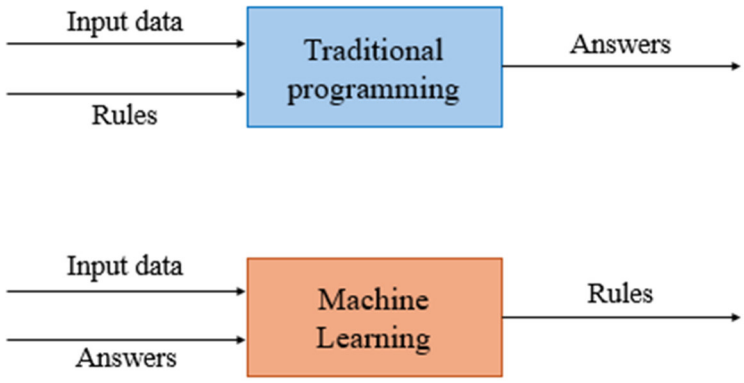


Figure 2 – Traditional programming vs. Deep Learning (TensorFlow, 2022)

The approach of Deep Learning can be advantageously used in Computer Vision. In the real world, objects that are supposed to be detected can have a lot of appearances. For example, when vehicles are to be detected, there are so many types and models, colours, customisations

of the vehicles in the world. Moreover, the camera „sees“ the vehicles always from different angles, distances, under different conditions. It is generally therefore impossible to define objects absolutely in terms of traditional programming. The better way is to use the Machine Learning approach and tell the computer how the objects usually look like. By the process of training (there is also online and offline training) the computer is taught about objects of interest, its shapes, common aspect ratios, dimensions, it also learns some typical features of it etc. After training the detection model on some dataset that contains a sufficient amount of training pictures (or videos), the computer is able to detect objects of such class with particular confidence and accuracy. (Chai et al., 2021 and Voulodimos et al., 2018)

Deep Learning methods basically use Convolutional Neural Networks (CNNs). The design of CNNs was inspired by neurons in human brain. CNNs have many layers inside and their standard conception is displayed in the Figure 3.

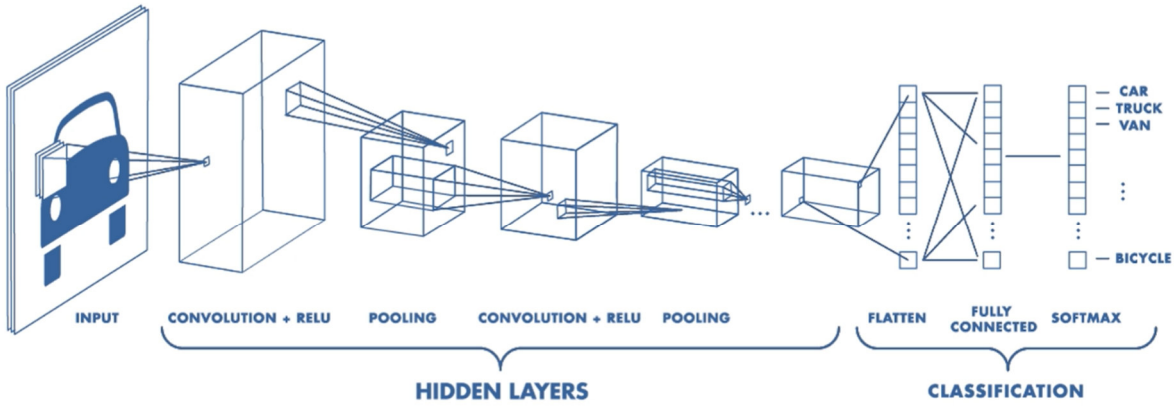


Figure 3 – Standard conception of the CNN (MathWorks, 2017)

Firstly, the input image is divided into small regions. These regions are provided as an input to the convolutional layer, where they are transformed by defined operation in order to emphasize some typical features of the object. Then the transformed region is provided to a pooling layer, where it comes to reduction of its resolution while maintaining the emphasized parts. These operations are repeated multiple times and at the end all the significant features of the object are extracted. Then comes the phase of classification, when the extracted and flattened features of the object are compared to the features obtained during the training phase (ground truth) in fully connected layers. The output of CNN is then a vector describing the affiliation of the object to each object class. (Chatterjee, 2019)

In the following text selected DL methods of automatic object detection will be briefly described. The main criteria to select DL methods for this thesis purposes were the accuracy and the speed of the algorithm.

Nowadays there are four main streams of DL detection methods:

1. R-CNN,
2. Single Look algorithms,
3. Transformer-based algorithms,
4. Diffusion/foundation models.

Their overview with description will be stated below.

2.1.1.1. R-CNN algorithms

As a standard CNN is unsuitable for automatic object detection due to a huge number of proposed Regions of Interest (RoIs), customized algorithms based on CNN have been developed. Generally, CNNs tend to be very accurate, but slower than alternative algorithms when passing data through them multiple times. The main effort is to find a way to maintain their accuracy and increase their speed.

R-CNN

R-CNN is an abbreviation of Region-based CNN and before it passes an image to CNN, it performs a selective search of RoIs. The selective search consists of sub-segmentation of the input image e.g., with Felzenwalb's algorithm complemented with a greedy algorithm used to reduce the amount of RoIs. Then 2 000 RoIs are warped, because a fixed size of RoI is needed and are proposed to the CNN. Then features are extracted from the CNN and passed to the Support Vector Machine (SVM) to separate different groups of features from each other.

The main drawback of this approach is the need to pass 2 000 RoIs through CNN, which is very time consuming. Due to its low speed, it cannot be used as a real-time detector. Moreover, the selective search phase cannot be taught by DL, which makes it less flexible and efficient. (Uijlings et al., 2013; Felzenszwalb and Huttenlocher, 2004; Girshick et al., 2014)

Fast R-CNN

In this algorithm an R-CNN method is made much faster. Only the input image is fed to the CNN unlike 2 000 RoIs in the R-CNN. CNN then extracts the feature map and identifies the region of proposals (RoI pooling is performed in just one layer). Then the process is similar with others – RoI pooling layer is passed to the fully connected layer, objects are classified with the softmax algorithm and bounding boxes are proposed. (Girshick, 2015)

Faster R-CNN

As the name indicates, Faster R-CNN is even Faster than Fast R-CNN. The architecture is very similar, but between the CNN and the RoI pooling layer is another small CNN called „*Regional Proposal Network*“, that proposes RoIs on the basis of DL. (Ren et al., 2015)

Mask R-CNN

This algorithm is an extension of the Faster R-CNN. Except standard detection, classification and bounding box creation it is able to define each object with the high precision to the individual pixels with the help of a binary mask. The architecture is similar, but instead of using RoI Pool by RoI pooling layer, Mask R-CNN uses RoI Align that re-aligns and re-scales each RoI to be passed to Mask Head, which makes the binary mask. There is even an extension for better precision called PointRend for smoothed and more accurate edges. (He et al., 2017)

R-FCN

The abbreviation R-FCN stands for Region-based Fully Convolutional Network. The main idea by this approach is that the whole object is predicted and detected by detecting its selected parts. For example, when a face of a human is divided to a 3x3 grid, in the left-hand top corner will be the right eye, in the centre will be the nose etc. With the appropriate dataset the model can be taught how these parts of the human face look like and what is their common mutual position. Then nine region-based feature maps are created, when each one will be detected in one cell of the grid. By this, votes will be obtained for each cell, how much it is a particular part of the face. Then a mean of these votes is obtained from the whole grid and determines the object class. (Dai et al., 2016)

2.1.1.2. Single Look algorithms

The algorithms that detect objects by looking only once on the input image will be stated in the following text.

SSD

The Single Shot Detector (SSD) firstly passes the input image through the CNN that produces multiple feature maps of different scales. The fully connected network is removed here, so the output of the CNN are the feature maps. Then a Detection Head generates bounding boxes of different scales and aspect ratios based on proposals of feature maps. These bounding boxes then slide across the image and search for the match with detected features. In order to remove duplications, the Non-Maximum Suppression (NMS) algorithm maintains only the bounding box with the highest Intersection of Union (IoU) and others are dropped. The SSD

algorithm is known to be very fast, it can reach 34 fps at the overall accuracy 59 % at the UA-DETRAC dataset (large dataset of traffic videos). (Liu et al., 2016; Zhao et al., 2021)

YOLO

The name YOLO is the abbreviation of You Only Look One, which suggests the main principle of this algorithm. The image is fed to the CNN which extracts the feature map and classifies the objects. YOLO then uses the sliding window approach, which was made effective by using CNN, because CNN can significantly reduce the resolution of the parts of the image while maintaining the features. For the detection of multiple objects, YOLO divides the image into a grid when each cell is treated as a separate image. Originally it was possible to detect only one object per cell, but this drawback was handled by the further versions by implementing anchor boxes for individual classes.

YOLO generally has many versions, each improving the code in some way. From the first YOLO in 2016 until nowadays there were for example following releases: YOLOv2, YOLOv3, YOLOv4, YOLOv5, PP-YOLO, Scaled YOLOv4, PP-YOLOv2, YOLOv6, YOLOv7 etc. (Jiang et al., 2022; Arya and Rawat, 2020).

2.1.1.3. Transformer-based algorithms

Transformer-based detectors represent a new generation of object detection algorithms inspired by the transformer architecture originally introduced for natural language processing (Vaswani et al., 2017). These models process images holistically using attention mechanisms, which enables them to model global relationships between parts of an image – something convolutional networks typically struggle with due to their local receptive fields.

Unlike classical methods based on anchor boxes and sliding windows, transformer-based detectors often frame object detection as a set prediction problem, where the network directly predicts a set of bounding boxes and class labels in a one-to-one matching fashion.

DETR

The DETR stands for DEtection TRansformer and was the first transformer-based object detector, introducing the idea of viewing object detection as a direct set prediction problem. It replaces hand-designed components like anchor generation and NMS with a transformer encoder-decoder architecture. DETR takes image features from a CNN backbone (typically ResNet) and feeds them into a transformer that learns to match these features to a fixed number of object queries. It achieves high accuracy, but one major limitation is its slow convergence and limited performance on small objects, which later models aim to address. (Carion et al., 2020)

Deformable DETR

Deformable DETR improves the original DETR by introducing multi-scale deformable attention, which focuses only on a small set of key sampling points rather than the full image grid. This makes the model much faster to train and improves detection performance on small and densely packed objects. Deformable DETR also enables multi-scale feature aggregation, making it more practical for real-world applications and closing the gap with traditional CNN-based detectors in terms of speed and accuracy. (Zhu et al., 2021)

DINO

DINO builds upon Deformable DETR (abbreviation DINO means DETR with Improved DeNoising and anchor boxes) and introduces denoising training and better initialization strategies for the decoder queries. These improvements lead to faster convergence, more stable training, and state-of-the-art accuracy on benchmarks like COCO (Common Objects in Context). DINO is currently one of the best-performing end-to-end object detectors, and it's widely used as a strong baseline for research in this area. It shows that transformer-based models can not only match but outperform CNN-based architectures in practical object detection tasks. (Zhang et al., 2022)

Grounding DINO

Grounding DINO extends the DINO model for open-set object detection using natural language. It incorporates text embeddings from large language models (like CLIP) and can detect objects described by free-form text prompts – even if the object class was not seen during training. This model is part of a growing trend of multi-modal transformers, which fuse vision and language, enabling powerful tasks like zero-shot detection or referring expression comprehension. It's widely used in conjunction with models like SAM (Segment Anything Model) for multi-purpose vision tasks. (Liu et al., 2023)

2.1.1.4. Diffusion/foundation models

In recent years, a new generation of object detection models has emerged that differ significantly from traditional approaches. Rather than relying on pre-defined rules or fixed structures, diffusion models and foundation models use highly flexible, data-driven techniques that allow them to understand and detect objects in much broader and more intelligent ways.

Diffusion-based models approach object detection as a step-by-step refinement process. They begin with a completely random guess about where objects might be located and then gradually improve these guesses through a series of small adjustments – much like how a blurry image becomes clearer when you slowly sharpen it. This method is inspired by image

generation techniques and allows the model to detect objects more creatively and flexibly, especially in complex scenes. However, these models are still quite new and can be slower or more demanding to run compared to older methods.

On the other hand, foundation models are very large, general-purpose AI systems trained on massive amounts of image and text data. What makes them unique is their ability to understand both images and language at the same time. These models are incredibly versatile and allow for tasks like open-vocabulary detection, where the model can recognize objects that were never explicitly part of its training. The main challenge with foundation models is their size, complexity, and the need for powerful hardware, which can make them difficult to deploy in smaller or real-time systems.

DiffusionDet

DiffusionDet reframes object detection as a denoising diffusion process, where object bounding boxes are treated as variables sampled and refined over multiple steps. Starting from a set of random boxes, the model iteratively refines them using learned denoising steps until accurate detections emerge. This generative approach enables detection without the need for predefined anchor boxes or proposal networks. While it's computationally more expensive than standard detectors, it demonstrates robustness, flexibility, and state-of-the-art performance, particularly for crowded scenes and ambiguous object boundaries. (Chen Z. et al., 2023)

OWL-ViT

OWL-ViT (Open-World Localization with Vision Transformers) is one of the earliest models to combine vision transformers with open-vocabulary capabilities, enabling the model to detect objects using text queries. It avoids the need for explicit class labels in training and performs well in zero-shot detection settings. Its architecture builds on ViT (Vision Transformer) and uses contrastive learning between image patches and text embeddings. This makes OWL-ViT highly flexible and capable of detecting novel object categories based purely on description – paving the way for human-interpretable, language-driven computer vision systems. (Minderer et al., 2022)

SAM

Although SAM (Segment Anything Model) is primarily a segmentation foundation model, it plays a crucial role in modern detection pipelines, especially when combined with models like Grounding DINO. SAM enables interactive, prompt-based segmentation of arbitrary regions and is trained on a massive dataset of masks. In object detection pipelines, SAM often follows the object localization stage (e.g., via Grounding DINO), providing pixel-level

precision for any detected region. It's part of the move towards unified vision models capable of detection, segmentation, tracking, and more – all in a single framework.

3.2. Automatic tracking of road users

The detection itself can find objects in the image, classify them, and provide their location. It is possible to count objects of the same class, draw a bounding box around them etc. In order to analyse their motion and kinematic quantities as well as their mutual interaction, it is necessary to keep track of each object.

Tracking accuracy metrics

Two metrics were established to evaluate the accuracy of tracking: *MOTA* (Multiple Object Tracking Accuracy) and *MOTP* (Multiple Object Tracking Precision). The following formulas 3, 4 represent the calculation. (Bernardin et al., 2006)

$$MOTP = \frac{\sum_{t,i} d_{t,i}}{\sum_t c_t} \tag{3}$$

$$MOTA = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t} \tag{4}$$

where: $d_{i,t}$ distance between the object and its predicted location,
c_t number of matches in the time t,
m_t number of misses in the time t,
fp_t number of False Positives in the time t,
mme_t number of mismatches in the time t,
g_t number of objects present in the scene in the time t.

There are several criteria under which the tracking methods can be classified (Figure 4).

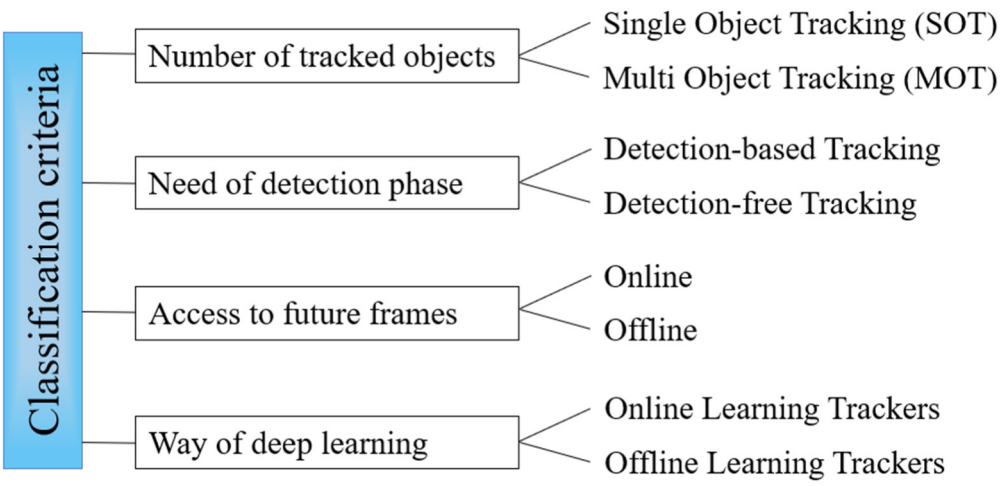


Figure 4 – Classification of tracking methods (Adžemović, 2025)

The first classification depends on the number of objects to be tracked. Single Object Trackers (SOT) can track only one object at the time, while Multi Object Trackers (MOT) track more objects at the same time. For the automatic traffic analysis purposes, the MOTs are mostly used.

According to the second criterion, tracking methods can be divided into two groups depending on the use of a detector. In case the tracker doesn't use the detector, the user must manually determine the object of interest that has to be tracked. Detection-based trackers make use of one of the detectors described in the previous chapter.

The third criterion divides the methods on the way of making predictions of the future position of objects. Online trackers make predictions immediately and from the motion history because it doesn't know the future position, which is the case of real-time tracking. Offline trackers can also use future frames of video record to optimize the predictions.

The last classification considers the way of deep learning of the tracking model. Trackers from the category of Online Learning Trackers learn about the particular object from the initial and subsequent frames which makes this kind of trackers more flexible. Offline Learning Trackers are trained offline only with the help of a dataset. (Klinger; Barla, 2022)

In the following text selected trackers will be briefly described.

GOTURN

Generic Object Tracking Using Regression Networks (GOTURN) is a CNN-based tracker that learns offline from a large dataset, which enables it to reach the speed up to 100 fps. The tracker uses two consequent frames as an input. These frames are fed into CNN (CaffeNet in

this case) and based on the prediction of this pre-trained CNN a new position of the object is predicted. (Held et al., 2016)

MDNet

The abbreviation MDNet stands for Multi-Domain Network. This tracker learns online. In order to maintain a sufficient speed of the tracker, in MDNet only a few last layers learn online, and the rest is pre-trained and works as a feature extractor. MDNet also rearranges the network into two parts. The first one is common for all domains (analysed videos), while the other is independent for each domain, which makes it more generic and flexible. (Jung et al., 2018)

ROLO

Recurrent YOLO (ROLO) uses the object detector YOLO combined with a Long Short Term Memory (LSTM) for tracking the object. It learns online and is designed to track single objects. The working principle of YOLO was explained earlier and YOLO in this case helps the tracker to focus on particular visual elements. The LSTM keep track of the spatio-temporal history and predicts the future position. When the YOLO fails to detect the object due to occlusion or blur, the LSTM can maintain the information and keeps the tracking stable. (Ning et al., 2017)

SORT + modifications

Simple Online and Realtime Tracking (SORT) represents an online detection-based tracker. The detection, which is the first phase, can be done by any detector (authors propose Faster R-CNN). Then the future position is estimated by linear constant velocity model and Kalman's filter. Then each proposed new position of objects is associated to some existing object, or a new object is created. The association is performed by Hungarian algorithm. (Bewley et al., 2016)

To improve SORT's accuracy and speed, there have been developed several modifications. Often used modification is DeepSORT, which uses a simple CNN to learn about the appearance of each object, that can be then easily identified after an occlusion. Hence, DeepSORT isn't only based on the objects' motion and velocity, but also on their appearance. Other extensions are e.g., StrongSORT, BoT-SORT, ByteTrack etc. (Wojke et al., 2017)

DEFT

DEFT is the abbreviation of Detection Embeddings for Tracking, which represents a very accurate multi-object tracker, that can draw not only 2D bounding boxes around tracked objects but also bounding boxes in 3D considering the depth of the object. The high accuracy and speed are reached by using intermediate feature maps from the object detector in order to use them in

the future frames to associate proposals with objects. The final association is performed with the help of Hungarian algorithm. The tracking algorithm is complemented with LSTM to predict physically possible future trajectories. (Chaabane et al., 2021)

FairMOT

The FairMOT's backbone is based on enhanced CNN, that can skip connections between low-level and high-level features and is able to solve issues with object alignment and different sizes. Then it uses a heat map for object centres detection and segmentation. In the next step the algorithm generates re-ID features, that are based on a trained model and helps to find the object after an occlusion. The main tracking is done by Kalman's filter and IoU method and the association is performed by the Hungarian algorithm as well as in SORT method. (Zhang et al., 2021)

CenterTrack

In CenterTrack algorithm objects, that are supposed to be tracked, are represented as points, which form the centre of their bounding box. The association of proposed object with the known object is performed by the greedy algorithm as the code takes the information about the past position of objects. The fact, that the objects are replaced by points simplifies the association, that can be reduced to displacement prediction and optical flow. The overall constellation of objects is projected to a heatmap. The main advantage is its speed due to its simplicity. On the other hand, the algorithm cannot handle longer occlusion as it takes only two consecutive frames. (Zhou et al., 2020)

ByteTrack

A real-time multi-object tracker that improves on SORT-like frameworks by associating not only high-confidence but also low-confidence detections. This reduces false negatives and preserves identities through occlusions without heavy Re-ID models. It achieves a very good ration between speed and accuracy and is widely adopted in MOT competitions. (Zhang et al., 2022)

TrackFormer

A transformer-based approach to multi-object tracking, extending DETR by introducing persistent "track queries" that follow objects over time. This allows the model to learn detection and association jointly, capturing long-term dependencies and complex interactions between objects in crowded scenes. (Meinhardt et al., 2022)

BoT-SORT

Combines a strong YOLOx detector, a robust Re-ID module, and advanced association strategies to deliver state-of-the-art accuracy. It leverages both motion and appearance cues effectively, making it reliable in crowded, fast-changing environments. (Aharon, N. et al., 2022)

ByteTrack 3D (AB3DMOT / 3D Extension)

An adaptation of the ByteTrack approach to 3D tracking using LiDAR or stereo data. It tracks objects directly in 3D space using motion models and low-confidence detection association, which is particularly valuable in autonomous driving and robotics scenarios. (Weng et al., 2020)

3.3. Surveillance camera calibration

When all objects of interest are successfully detected and tracked, the process of obtaining desired quantities can be initiated. In the basic line it is possible in this point to determine the position of objects in the image [x, y] and through the change of their position in time to calculate velocity, acceleration or deceleration, direction vector etc. Nevertheless, the quantities are expressed related to the pixels e.g., velocity units are pixels/sec etc.

The next phase is the process called camera calibration when a transformation matrix between points in pixel coordinates and real-world coordinates is determined. The main calibration process will be described on a pinhole camera model, which is commonly used and simplest model of the camera. The pinhole camera model is displayed in the Figure 5.

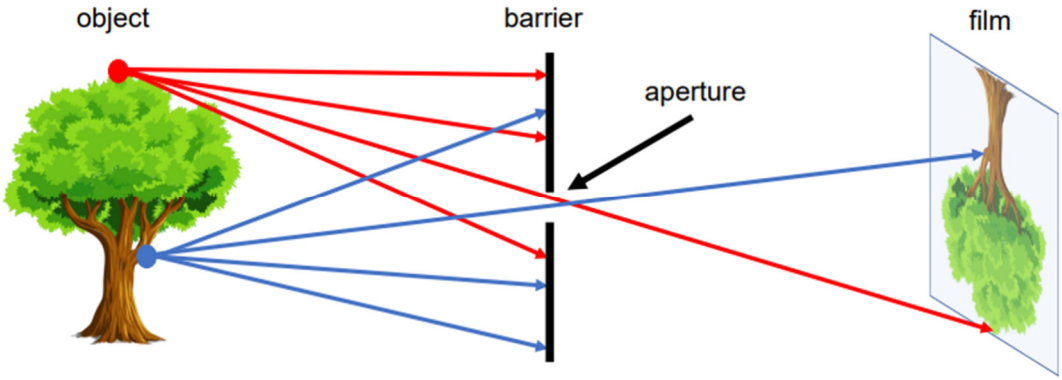


Figure 5 – A pinhole camera model (Hata and Savarese)

For the calibration purposes of traffic surveillance cameras, the authors have developed various methods. These can be generally divided into two main groups:

a) PnP (Perspective-n-Point),
b) VP (Vanishing Point acquisition).

3.3.1. PnP calibration method

Method called Perspective-n-Point is based on the knowledge of coordinates of several points in the real world and corresponding coordinates in the image. The main goal is to determine a transformation matrix which can robustly describe the relation between the image and real-world points. The transformation matrix consists of parameters of an intrinsic and extrinsic matrices of the camera which will be described further in this chapter. The minimal number of points needed for the calibration is based on the assumptions made by authors, but generally, when no calibration parameter is neglected, at least six non-colinear points are needed in the 3D coordinate system. Most authors assume the road as planar, which results in the decrease of the minimal number of points to four (z-world coordinate of each point is assumed zero or constant).

The camera intrinsic matrix *K* consists of the parameters of this configuration, such as focal length (distance between the barrier and the image plane – film), camera centre (coordinates of the pinhole) and skew (angle of image axis).

The following formula describes the way of coordinates transformation from the pixel coordinate system to the camera coordinate system.

$$Image\ coordinates\ (u, v) = K \cdot Camera\ coordinates\ (x_c, y_c, z_c) \tag{5}$$

Where the K stands for an intrinsic matrix describing the internal geometry and optical properties of the camera:

$$K = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

The parameters are f_x, f_y the focal lengths in pixel units, s is the skew coefficient and c_x, c_y are coordinates of the centre point of the camera.

Then a rotation (angles of the world axes in the camera coordinates) *R* and translation (camera coordinates of the world origin) *T* matrix are formed in order to obtain the transformation to the object's world coordinates. This matrix is called camera extrinsic matrix,

and its values change with the change of the camera's position. The transformation formula is as follows:

$$Camera\ coordinates\ (x_c, y_c, z_c) = [T, R] \cdot World\ coordinates\ (x_w, y_w, z_w) \quad (7)$$

Then combined to obtain a formula of transformation between image coordinates and world coordinates:

$$Image\ coordinates\ (u, v) = [T, R] \cdot K \cdot World\ coordinates\ (x_w, y_w, z_w) \quad (8)$$

Assuming for the purpose of the thesis only camera recordings with a road considered as a planar surface will be used, the parameter z_w can be put as a zero-height level, hence $z_w = 0$.

This way a planar homography is obtained:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = H \begin{bmatrix} x_w \\ y_w \\ 1 \end{bmatrix} \quad (9)$$

Where H is a 3x3 homography matrix combining unknown parameters from the intrinsic and extrinsic matrix.

After algebraic adjustments of the formula above, each point in a pixel form can be defined as:

$$u = \frac{h_{11}x_w + h_{12}y_w + h_{13}}{h_{31}x_w + h_{32}y_w + h_{33}} \quad (10)$$

$$v = \frac{h_{21}x_w + h_{22}y_w + h_{23}}{h_{31}x_w + h_{32}y_w + h_{33}} \quad (11)$$

Which can be rearranged to:

$$x_w h_{11} + y_w h_{12} + h_{13} - u(x_w h_{31} + y_w h_{32} + h_{33}) = 0 \quad (12)$$

$$x_w h_{21} + y_w h_{22} + h_{23} - v(x_w h_{31} + y_w h_{32} + h_{33}) = 0 \quad (13)$$

As can be seen from the above two equations, a solution with 9 unknown parameters must be found. However, the last component in the homography matrix h_{33} stands for a scale and can be considered = 1. Considering the assumption that z-coordinate is 0 as the ground is assumed

plain, at least 4 corresponding points (each point has two coordinates → 8 known parameters) between the image and the real world have to be provided to find the solution of the system of equations.

Then an equation in form of (14) will be obtained.

$$A\vec{x} = \vec{0} \tag{14}$$

In this case, A is the matrix of known values of zeros, ones and combinations of known coordinates of points in pixel and world coordinate system and $\vec{x}$ is the flattened matrix of unknown coefficients. The calculation of these coefficients is approximate and based on minimizing the algebraic error, which leads to the problem of eigenvectors.

For the input known points there are several rules that must be kept for convergence. Basically, it is advantageous to provide more than four known points. These points also shouldn't be colinear to maintain the independence of the points. After obtaining the transformation matrix, it is necessary to perform a non-linear optimization using a gradient descent algorithm due to a geometric error. (Xu and Wang, 2012; Krishna, 2022)

The first estimation of the homography matrix can be found using python in two ways. One is programming exact equations as described above. The other is using a built-in function called cv.findHomography, where it can be chosen if the solution is found using e.g. least square method, RANSAC method etc. Generally, both methods mostly lead to a solid first estimation which can be used further. (Galliot, 2022)

After obtaining the first estimation of the solution, a sum of errors is computed. In order to define the error, the real-world points coordinates are subtracted from the projected real-world coordinates, which are projections of the image plane points using the homography matrix. A possible outcome for four points is displayed in Figure 6.

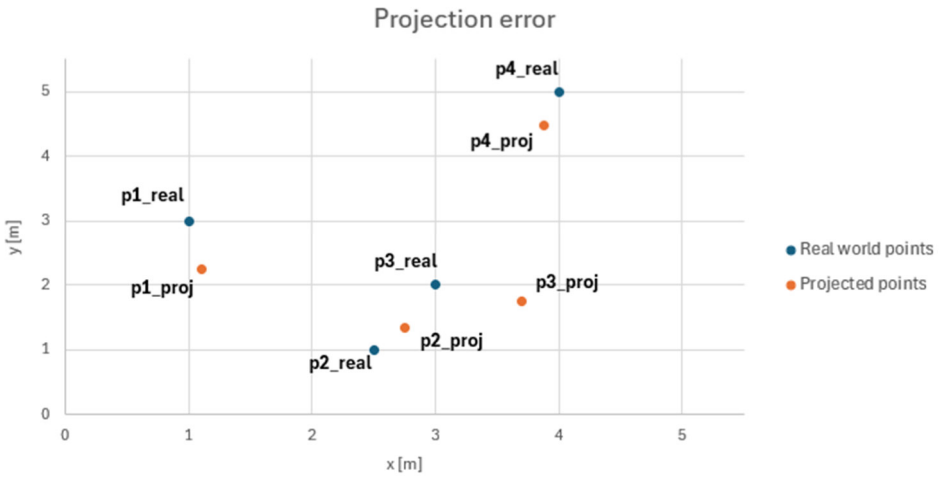


Figure 6 – Projection error (author)

To define the error of the projection, the difference between corresponding point is calculated and summed up, following the formula below.

$$err = \sum_{i=1}^{n} \sqrt{\Delta x_i^2 + \Delta y_i^2} \tag{15}$$

At this point the main goal is to minimize the error, which means the difference between the ground-truth and projected points. This can be done with a built-in function from python from *scipy* library called scipy.optimize.minimize. As an argument the error is given and after several iterations, the function optimizes the homography matrix so that the error is minimal. The precision often reaches several centimetres of error what is acceptable for the purposes of this thesis (Figure 7).

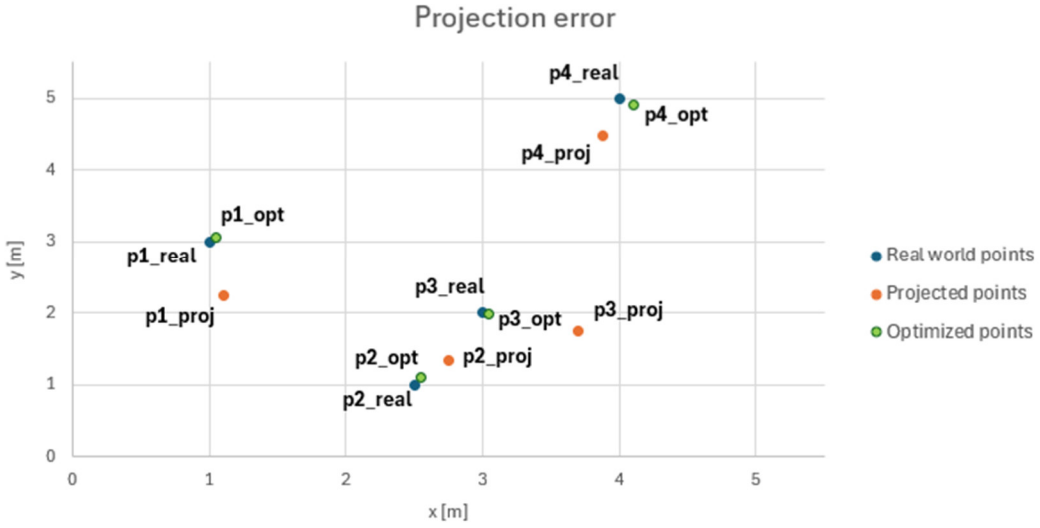


Figure 7 – Optimized homography matrix (author)

In this thesis, the ground truth points were obtained by a GNSS total station measurements physically at the crossing. However, the PnP method is versatile and for obtaining the ground truth points, an aerial picture with scale can be an alternative. The images in a sufficient quality, that can be very closely zoomed in are aerial pictures from the cadastral map, which are available on the internet. Moreover, distances can be measured precisely up to two decimal places on this map. An example of such aerial image with measurements (blue lines) is displayed in the Figure 8.

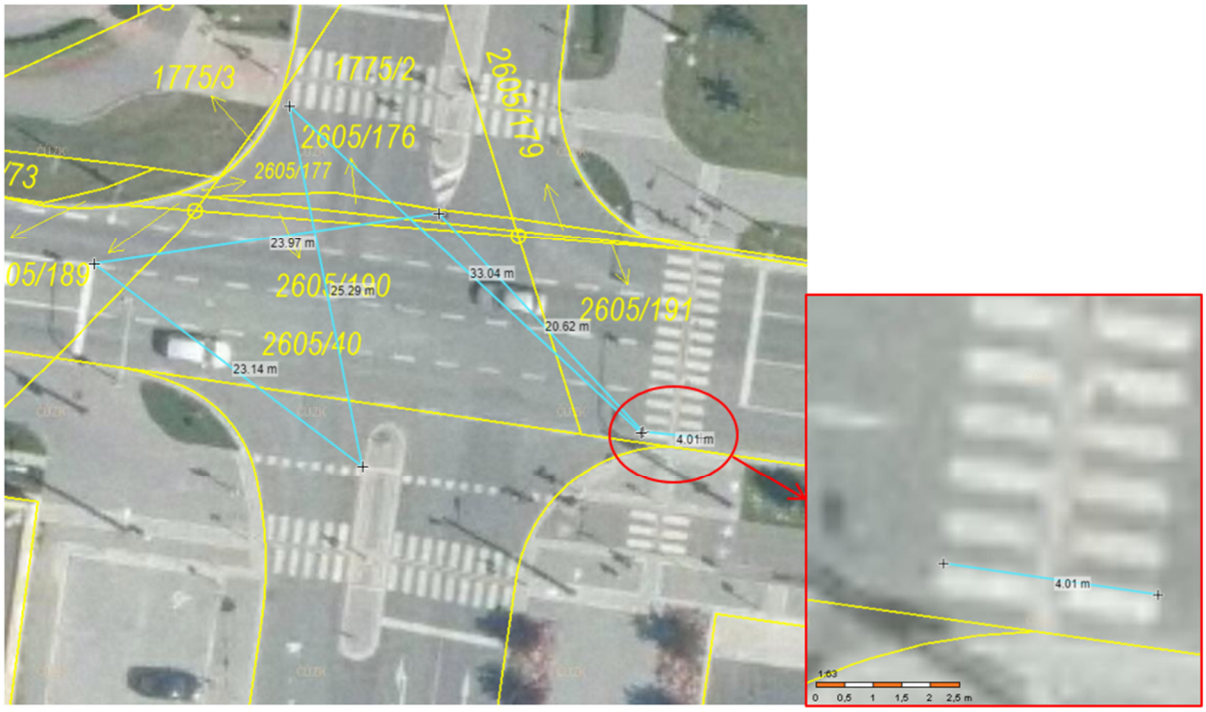


Figure 8 – Measuring distances on a cadastral map (author on maps from ČÚZK)

In the right-hand figure a measurement of pedestrian crossing marking lane's length is represented. Such standardized markings and solid objects can be advantageously used for obtaining the ground truth due to their standardized dimensions.

It is necessary to determine how precisely it is possible to measure certain distances in the aerial images of the cadastral maps. The advantages are the facts, that the images can be closely zoomed in, and the maps enable to measure with the precision on two decimal places. Also, standardized solid objects and markings can be used to determine the ground truth distances. On the other hand, the main error occurs from the aiming at individual points by hand by the mouse. Then, in the image from the camera, the user has to click on the exactly same place to determine the corresponding ground truth point, which can cause some error due to the perspective distortion.

Related papers

The authors in Song et al. (2022) solve a cross-camera calibration having multiple surveillance cameras above a road in a row with a little overlap. The calibration method of calibrating individual cameras is the PnP method adding the correction of both tangent and radial distortion. No errors or more details on calibrating individual cameras were provided.

In a paper by Tang, Z. et al. (2019) the authors assume the earth as a flat surface and use the correspondence between 5-14 2D points in the image and their 3D world coordinates known from Google Maps. For the cost function a reprojection error in pixels is used and the optimization is performed by the least squares or RANSAC method. The only error concerning the calibration stated by the authors is the reprojection error of 11.52 pixels mainly due to the inaccuracy of the Google Maps. When there is a radial distortion in an image the straightening of the curved traffic lanes is firstly performed.

Eddy et al. (2018) solved the camera calibration concerning the measurement of a cyclist motion. They mounted the cameras on the top edge of a truck to obtain a top-down view and thus to minimize the lateral errors. The intrinsic camera parameters are considered known, and the barrel distortion is solved by a quadratic function. The cyclist moves on a grid drawn on a road of known dimensions and this way the correspondences between 2D image points and 3D world points are obtained. An absolute error of ± 3 pixels meaning 4 cm in the world coordinates was obtained.

In Gunawan et al. (2019) the authors allow user to insert the known distances between points manually or to specify the number of known points to form parallel lines. Then, an analysis of the optimal number of known points' correspondences was performed with the general conclusion that the more known points are used the better accuracy can be obtained. For the Direct Linear Transformation algorithm evaluation purpose, the authors conducted a controlled experiment with toy cars and drawn roads under various camera angles and heights. Only four points are used, namely the upper left, upper right, lower left and lower right point. The authors obtained the accuracy of 0.41 $m \cdot s^{-1}$, which means 3.86 % relative error and by using background subtraction detection method an Euclidean distance error of 12.07 pixels.

In the paper by Zhu et al. (2022) the authors developed a method to detect vehicles by a transformation of the surveillance camera point of view to bird eye's view. This way authors take the advantage that in the bird eye's view the scale of the vehicle is consistent, and no vehicle's bounding boxes overlap. They assume the ground as planar and no non-linear camera distortion. As the ground is assumed planar, the homography is used to transform coordinates

between two planes. To obtain the bird eye's view of the particular area, the authors use Google Maps and match selected corresponding points together. The homography is then solved by DLT.

Feng et al. (2020) record the vehicle's movement from side perpendicular to the vehicle's longitudinal axis and in the 30° angle. To correct the lens distortion a checkerboard was used. They use three points in each frame as the reference points: rear hub back-end, rear hub front-end and front hub front-end. Then the coordinates of the points are extracted, and the transformation matrix is found by the DLT. The authors reached in the camera angle 90° with the longitudinal vehicle's axis the speed error of 1.0 %, acceleration error of 6.4 % and travel distance error of 1.9 %. When the camera angle with the longitudinal vehicle's axis was 30°, the speed error was 3.7 %, acceleration error 5.6 % and travel distance error 4.9 %.

The authors Huang et al. (2020) use the known pattern and closed form solution method proposed by Zhang (2000). However, instead of using the checkerboard, they use multiple pedestrian crossings forming a pattern of known length and width of the lines and their spacing reaching with their method a mean relative distance error of 0.21 m.

In the Luvizon et al. (2016) the authors assume the earth surface as planar and pinhole camera model. According to the planar ground assumption the transformation between world and image coordinates was reduced to plane-to-plane homography, when minimally 4 corresponding point were needed. They used markings on the road by the inductive loops forming a rectangle 4.8 m x 2.0 m. The error of speed measurement was evaluated as 4 % within the acceptable limits.

A fully automatic approach to camera calibration is presented in the publication Bhardwaj et al. (2018). The authors firstly detect vehicles by traditional background subtraction method and then they use Deep Neural Network based method on extracting feature points of the vehicles. Based on ten common sedan 3D models, they obtain the distances between the corresponding feature points. In the final stage filtering and averaging techniques are applied to eliminate outlying values. A distance RMSE (Root Mean Square Error) has an average value of 8.98 %.

Bartl et al. (2020) proposed another fully automatic camera calibration method similar to the method in Bhardwaj et al. (2018). The authors detect vehicles by the Faster R-CNN detector and classify them according to their make, model and year. Then a neural network localizes landmarks (e.g., wheels, licence plates, logos, corners) on the vehicles and obtain their mutual distances according to the nine most common 3D car models. To reduce the error caused by

outliers, the re-projection error is minimized by Differential Evolution minimizing. The RMSE error is evaluated in the same way as in the Bhardwaj et al. (2018) reaching 4.03 % while by using the AutoCalib on the same dataset, they obtained the RMSE of 6.56 %.

An alternative approach was introduced in the publication Koetsier et al. (2019), where the calibration is performed semi-automatically. As a video record, he uses a record of a camera, which parameters are unknown. He therefore calibrates it by defining totally 25 points on the video record (pedestrian crossings, road markings, trash bins, trees etc.) and measuring them using the aerial image of the region of interest. This principle is shown in the Figure 9. However, the authors don't state any calibration error.

Figure 9 – Camera calibration using points measured in an aerial image (Koetsier et al., 2019)

In the publication Juránek et al. (2019) the authors use manual calibration of the camera when automatically processing records of vehicles driving through a bend by measuring the physical distance between selected points. They also assume the road surface as planar. The homography matrix is evaluated by the least squares method and the authors reach the relative error of 1 %.

The following Table 1 represents the overview of papers concerning the camera calibration by the PnP method. The information about the way of obtaining the calibration data, degree of automation and precision (if stated) is provided.

Table 1 – Overview of PnP calibration methods, their precision and automation.

Author	Publication	Methods of Calibration based on:	Auto	Precision (if stated)
Juránek, R. et al. (2019)	Visual Analysis of Vehicle Trajectories for Determining Cross-Sectional Load Density.	Manual measurement of distance between determined points in a curve.	✗	Relative distance error = 1 %
Bhardwaj, R. et al. (2018)	AutoCalib: Automatic Traffic Camera Calibration at Scale.	Extracts key-points on detected vehicles and compares distances with 3D model. Each vehicle has multiple calibration values which eliminates outliers.	✓	Average RMSE of distance measurement = 8.98 %
Koetsier, C. et al. (2019)	Trajectory Extraction for Analysis of Unsafe Driving Behaviour.	Manually measures distances between selected points in an orthophoto.	✗	—
Bartl, V. et al. (2021)	Automatic camera calibration by landmarks on rigid objects.	Automatic detection of landmarks on detected vehicles and distance comparison between these landmarks on 3D model and detected vehicle.	✓	Average RMSE of distance measurement = 4.03 %
Luvizon, D. C. et al. (2016)	A video-based system for vehicle speed measurement in urban roadways.	Known distances in the real world.	✗	Absolute speed error: -0.5 $km \cdot h^{-1}$
Song et al. (2022)	Target Tracking and 3D Trajectory Reconstruction Based on Multicamera Calibration.	Radial and tangential distortion correction. Multiple cameras in a row with a little overlap.	✓	—
Tang, Z. et al. (2019)	Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification.	Radial distortion correction. Take 5-14 manually selected points from Google Maps. Optimization through the reprojection error.	✗	Reprojection error = 11.52 pixels
Eddy et al. (2018)	Camera-based measurement of cyclist motion.	Barrel distortion correction. Intrinsic parameters known, extrinsic obtained by points' correspondences on the grid drawn on the road.	✗	Absolute distance error = ± 3 pixels (i.e. 4 cm)

Gunawan et al. (2019)	Detection of vehicle position and speed using camera calibration and image projection methods.	User inserts known distances or selects the number of points forming parallel lines. They use the DLT algorithm to obtain the parameters.	✗	Relative speed error = 3.86 % (i.e. 0.41 $m \cdot s^{-1}$ = 1.5 $km \cdot h^{-1}$) Average distance error = 12.07 pixels
Zhu et al. (2022)	Monocular 3d vehicle detection using uncalibrated traffic cameras through homography.	Authors select corresponding points from Google Maps. Then calibrate the camera through homography by a DLT algorithm.	✗	—
Feng et al. (2020)	A Calculation Method for Vehicle Movement Reconstruction from Videos.	Distortion corrected with a checkerboard. Consider three corresponding points on the vehicle in each frame, use DLT algorithm.	✗	According to the angle of the camera with the longitudinal plane of the vehicle. 90° Relative distance error = 1.9 % Relative speed error = 1.0 % Relative acceleration error = 6.4 % 30° Relative distance error = 4.9 % Relative speed error = 3.7 % Relative acceleration error = 5.6 %
Huang et al. (2020)	Comparative analysis & modelling for riders' conflict avoidance behavior of E-bikes and bicycles at un-signalized intersections	Use known dimensions of the pedestrian crossing.	✗	Relative distance error = 0.21 m

3.3.2. Vanishing Point acquisition

Vanishing points are generally points in which parallel lines meet in the 2D projection due to the perspective and other distortions. The lines are parallel if their vanishing points lie on the so-called horizon line, which is the line that comprises individual vanishing points made from lines on the same plane.

The main challenge nowadays is to find vanishing points in the image view in an automatic way. In the following text, authors using various methods of detecting vanishing points will be stated.

Generally, there are three main streams of obtaining vanishing points:

- manually,
- automatically by detecting parallel lines,
- by a Convolutional Neural Network (CNN).

The first way of obtaining the vanishing points manually is the simplest solution. In this way the user sets the coordinates of the vanishing points by optically evaluating the intersections of various lines in the image that are in the real world considered to be parallel. However, nowadays authors aim to acquire the straight lines in the image automatically.

In order to determine the vanishing points automatically by traditional algorithms, authors use mostly some kind of edge detector (Canny, Sobel etc.) or tracker finding the vehicles' trajectories to emphasize the straight lines in the image heading to some vanishing point. Then, usually Hough transformation is used to collect lines of desired direction to determine vanishing points. Authors use very often longitudinal lines on the road or on detected vehicles or fragments of their trajectories to obtain the first vanishing point and the horizon. Then, they find the second vanishing point by identifying the lines in perpendicular way, e.g. rear or front side of the vehicles.

The state-of-the-art method uses an own CNN trained to find appropriate vanishing points. The CNNs outperform the previous algorithms and are able to detect any markings on the road, road lanes and vehicle trajectories with high accuracy. The CNNs are mostly trained to recognize the geometric shape and appearance of the road or road markings that very often aim to the first vanishing point.

Calibrating the camera using the found vanishing points

In this paper, a method of two vanishing points' calibration will be stated in order to describe the way of obtaining intrinsic and extrinsic parameters of the camera. Dividing the camera calibration by vanishing points into more categories is beyond the scope of this paper. The main sources of the description will be two comprehensive papers by Guillou et al. (2000) and Orghidan et al. (2012).

Consider an image containing an object of two sets of parallel lines that define two different vanishing points. A line passing through these two vanishing points is called the horizon line (green in the Figure 10). Authors also often make assumption that the skewness is zero, scale factor equal to one and that the principal point lies in the centre of the image. The projection of

a rectangle onto the image plane, while the rectangle and the image plane aren't perpendicular, and obtaining the two vanishing points *Fu* and *Fv* is displayed in the Figure 10.

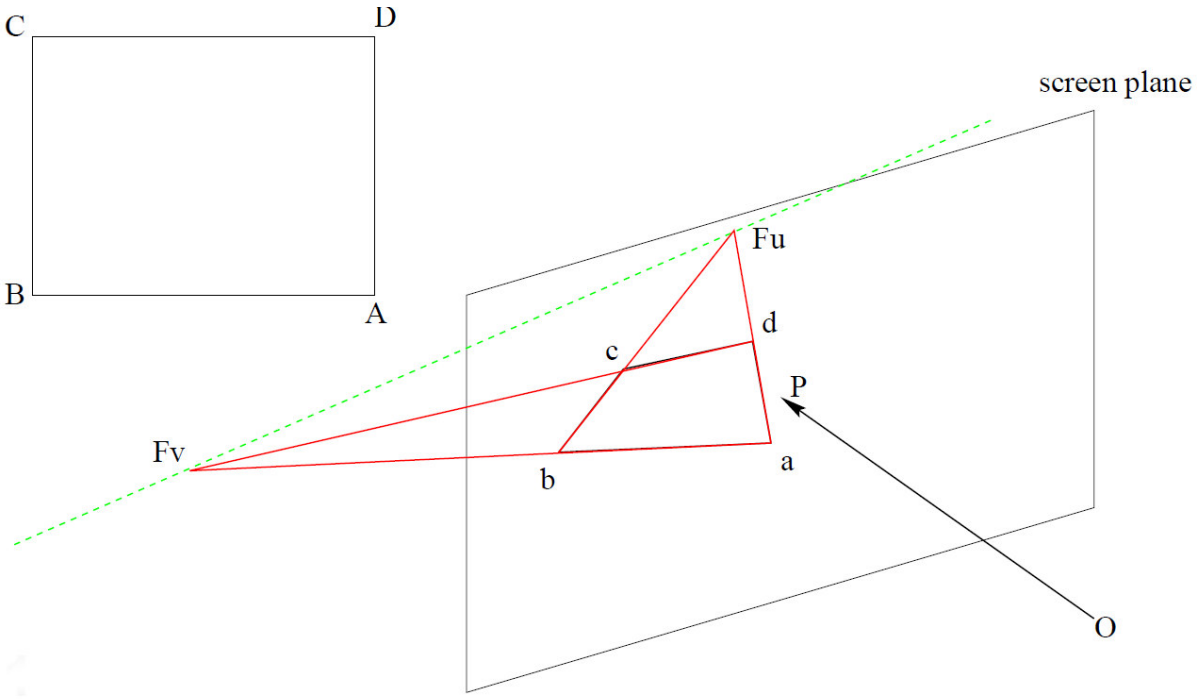


Figure 10 – A rectangle projection and obtaining of two vanishing points (Guillou et al., 2000)

In the Figure 3 the point *O* stands for the projection centre and the point *P* is its projection in the image plane. Supposing the orthogonal projection *Puv* of *P* on the horizon line.

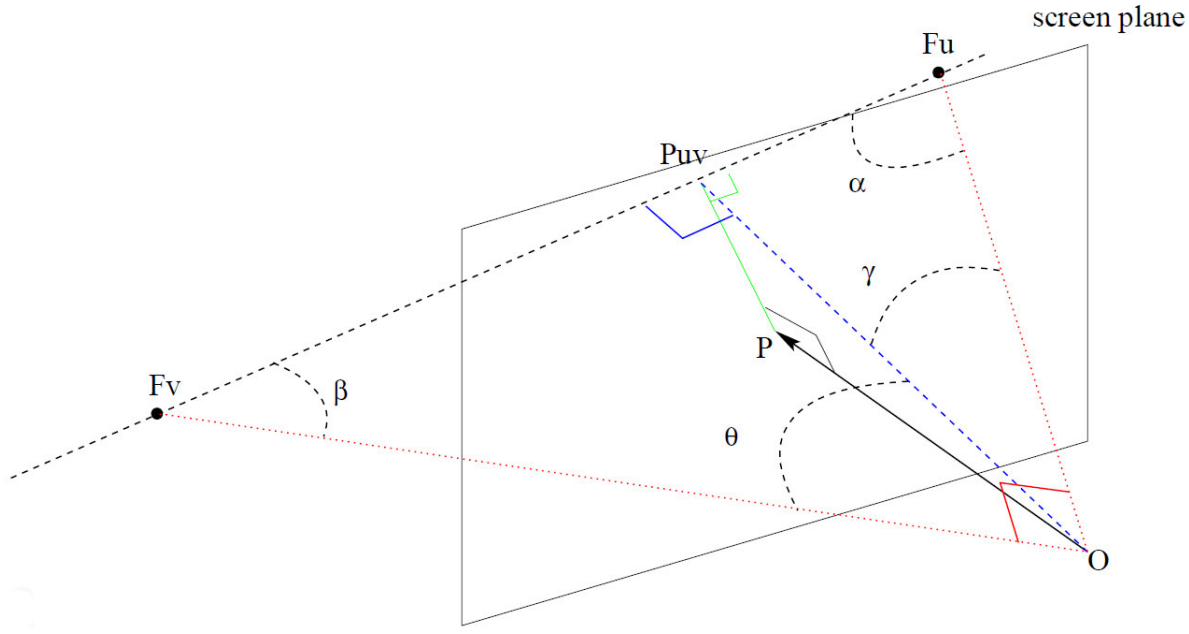


Figure 11 – Obtaining the focal length (Guillou et al., 2000)

To obtain the focal length, we compute:

$$f = |OP| = \sqrt{(OP_{uv})^2 - (PP_{uv})^2} \tag{16}$$

The length of $|PP_{uv}|$ is already known and $|OP_{uv}|$ can be calculated based on the right triangle's solution as:

$$|OP_{uv}| = \sqrt{|P_{uv}F_v| \cdot |P_{uv}F_u|} \tag{17}$$

After determining the intrinsic matrix, a rotation matrix is computed using the computed focal length and similarity between vectors $\vec{u}, \vec{v}$ and $\vec{w}$ determined in the rectangle in one of its vertices and corresponding vectors $\overrightarrow{u'}, \overrightarrow{v'}$ and $\overrightarrow{w'}$ in the centre of projection, while $\overrightarrow{u'}$ and $\overrightarrow{v'}$ head to the vanishing points F_u and F_v. (Figure 12).

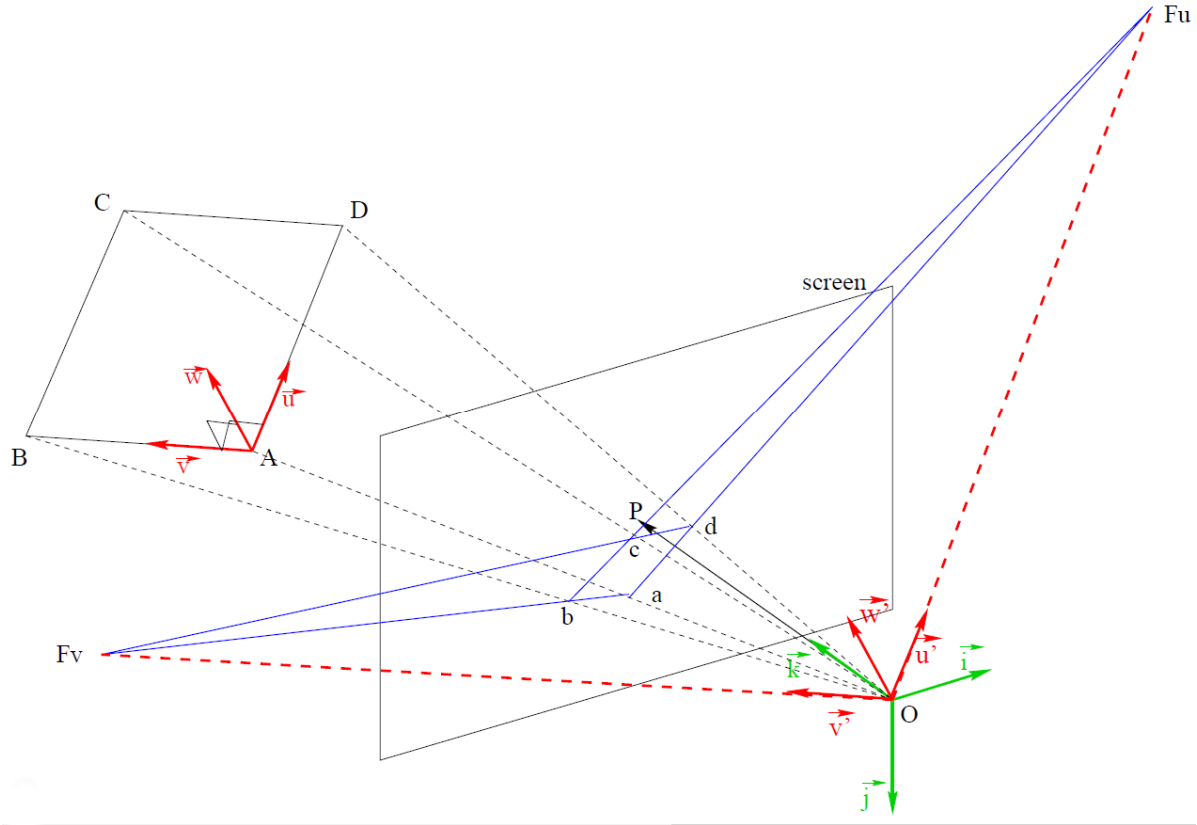


Figure 12 – Principle of obtaining the rotation matrix (Guillou et al., 2000)

Now, the rotation between the camera coordinate system $\vec{\imath}, \vec{\jmath}$ and $\vec{k}$ corresponds to the rotation between the camera coordinate system and the world coordinate system. The vectors $\overrightarrow{u'}, \overrightarrow{v'}$ and $\overrightarrow{w'}$ can be then expressed as:

$$\overrightarrow{u'} = \frac{\overrightarrow{OF_u}}{|\overrightarrow{OF_u}|} = \left(\frac{F_{ui}}{|\overrightarrow{OF_u}|}, \frac{F_{uj}}{|\overrightarrow{OF_u}|}, \frac{f}{|\overrightarrow{OF_u}|} \right) \tag{18}$$

$$\overrightarrow{v'} = \frac{\overrightarrow{OF_v}}{\left|\overrightarrow{OF_v}\right|} = \left(\frac{F_{vi}}{\left|\overrightarrow{OF_v}\right|}, \frac{F_{vj}}{\left|\overrightarrow{OF_v}\right|}, \frac{f}{\left|\overrightarrow{OF_v}\right|}\right) \tag{19}$$

$$\overrightarrow{w'} = \overrightarrow{u'} \times \overrightarrow{v'} \tag{20}$$

and the resulting rotation matrix R is:

$$R = \begin{pmatrix} \frac{F_{ui}}{\sqrt{F_{ui}^2 + F_{uj}^2 + f^2}} & \frac{F_{vi}}{\sqrt{F_{vi}^2 + F_{vj}^2 + f^2}} & w_i' \\ \frac{F_{uj}}{\sqrt{F_{ui}^2 + F_{uj}^2 + f^2}} & \frac{F_{vj}}{\sqrt{F_{vi}^2 + F_{vj}^2 + f^2}} & w_j' \\ \frac{f}{\sqrt{F_{ui}^2 + F_{uj}^2 + f^2}} & \frac{f}{\sqrt{F_{vi}^2 + F_{vj}^2 + f^2}} & w_k' \end{pmatrix} \tag{21}$$

The translation vector is then computed based on a segment e.g. $\overrightarrow{AP}$, which length is known. Points A' and P' are its perspective projection, point P'' lies in the intersection of the OP line and the parallel line with the AP line passing through the point A'. The whole configuration is displayed in the Figure 13.

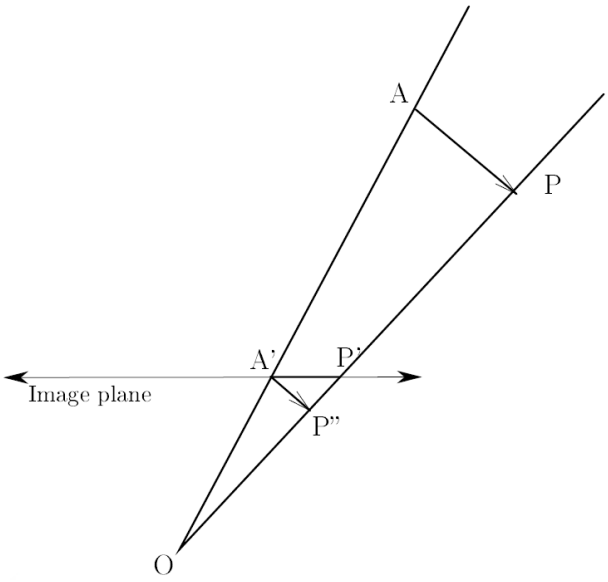


Figure 13 – The projection of the segment AP (Guillou et al., 2000)

As can be seen, the $\left|\overrightarrow{OA}\right|$ can be calculated from the principle of triangle's similarity as:

$$\left|\overrightarrow{OA}\right| = \frac{\left|\overrightarrow{OA'}\right| \cdot \left|\overrightarrow{AP}\right|}{\left|\overrightarrow{A'P''}\right|} \tag{22}$$

Hence, the translation vector will be obtained:

$$T = \left|\overrightarrow{OA}\right| \cdot \frac{\overrightarrow{OA'}}{\left|\overrightarrow{OA'}\right|} \tag{23}$$

Related papers

The authors in Miles et al. (2022) use the projected rectangle onto the road surface to find the two vanishing points. They assume the road surface as planar, so all points projected onto the road have their z-coordinate zero. The other assumption of the authors is neglecting the barrel distortion. A rectangle is then projected onto the road between two adjacent lines dividing individual road lanes, so the width of the rectangle is known. As the rectangle has always its opposite sides parallel, after projection the sides will be concurrent and so their intersections can be found and recognized as the vanishing points. The rectangle's width is considered by the software to calibrate the camera's scale.

In the Zhang et al. (2021) the authors designed a deep CNN called DeepCN to automatically calibrate the camera. They use the road markings' shapes, which are unique and repeat in different scenarios to find the parallel directions and thus the vanishing points. They taught the CNN the appearance of the road markings, obtained the vanishing points this way and thus calibrated the camera computing the calibration parameters. The DeepCN reaches the mean absolute error of 8.0 % in the measurement of the landmark's length and 6.1 % in the landmark's width on the dataset of the authors Zhang et al. The authors widened the use of the proposed algorithm also to automatically calibrate the camera using the DeepCN when the camera aims at the curved road and reached the mean relative error of the landmark's length and arc length of 7.3 % and the mean absolute error of the lane width of 3.6 %.

Another approach proposed Tang, Z. et al. (2019), who use moving human to estimate the vanishing points' position. They assume the ground as planar, and the height of the camera is known. The main assumption for using their method is recording at least one moving human captured in at least three different non-colinear positions. In order to find the vanishing points the authors find heads and feet of the humans as intersections of the main axis of the ellipsoid drawn around each human with their bounding box. Authors then assume the differences in humans' heights negligible. By connecting the main axes of humans' ellipsoids, the authors obtain the vertical vanishing point, by connecting the heads or feet of individual pairs of humans the lines converge on the horizon line in individual horizontal vanishing points.

The paper of Wu et al. (2020) solves the calibration of multiple cameras in a row with a little overlap. The authors make the assumptions of planar road, centre position of the principal point and the known height of the camera. The focal length and rotation parameters are obtained from the vanishing points. The first one is found as the intersection of the road lanes' lines. If the second, vertical vanishing point is difficult to be obtained, the authors use the known length of the landmarks on the road. The authors state the accuracy of the cross-camera calibration as a whole which can't be compared to the accuracy reached by the individual camera calibration.

Kocur and Ftáčnik (2021) use the CNN to obtain the vanishing points. They make the assumption of the zero skew, centre position of the principal point and no distortion. Firstly, they detect vehicles using the CenterNet object detector pre-trained on the MS-COCO dataset. These vehicles are cropped, resized, and put in the vanishing point detection network that produces heatmaps for each vanishing point. Based on the found vanishing points, the focal length is computed and thus the intrinsic camera matrix, horizon line and the normal of the road plane. The authors don't provide the scaling parameter which would need any real distance to be known or calculated and therefore they evaluate the error in the relative difference of ratios of two different measurements. They reached the mean error of 14.95 % on the BrnoCompSpeed dataset (straight highway) and 8.66 % on the BrnoCarPark dataset (parking lot).

In the Giannakeris et al. (2018) the authors use similar automatic approach as Dubská et al. (2014). They consider zero-pixel skew, square shaped pixels, and principal point of the camera in the centre as tolerable errors and assume the earth surface as planar. Therefore, all the points are supposed to lie on a ground plane and not in a 3D space. The first vanishing point's direction is the direction of the traffic stream which is acquired by using the Hough algorithm and KLT tracker to vote in the diamond space. The second vanishing point is also considered parallel to the road surface and perpendicular to the first vanishing point and is acquired by collecting the horizontal edges while excluding edges heading to the first vanishing point. Further the authors calculate the position of the third vanishing point, focal length and the road plane normal vector. To convert the normalized distances into the real, two 3D points of known world coordinates are used e.g., from the pre-made 3D vehicle models.

The authors Tang, X. et al. (2020) make use of knowing only one vanishing point complemented by the knowledge of the known lane width and length of the road marking. They assume that the principal point lies in the image centre. The vanishing point is acquired by the authors from the direction of the lines on the road lane's edges. The vehicle spatial relative error of 1.93 % is obtained with this method.

In Zwemer et al. (2022) the authors refer to Brouwers et al. (2016) when calibrating the camera. Both groups of authors find the first and second vanishing point, in Zwemer et al. (2022) according to the straight road lanes and in Brouwers et al. (2016) connecting heads and feet of detected persons. Then, the vertical and horizontal vanishing point's orientation is used to calibrate the camera.

In the paper by Kocur and Ftáčnik (2020) and Kocur (2019) the authors calibrate the camera according to Sochor et al. (2017) who enhance the automatic calibration method by Dubská et al. (2014) by more precise vanishing point's detection thank to 3D models of vehicles and thus obtaining the scene scale. With their methods of perspective transformation, they reach the mean speed error of 0.9 $km \cdot h^{-1}$.

The method proposed by Liu, T. and Liu, Y. (2021) is a combination of the PnP method and methods based on the acquisition of the vanishing point. Firstly, they manually select two pairs of parallel lines which are mutually orthogonal and thus obtain the first and the second vanishing point. Then, lengths of line segments are manually measured in the 3D world and used as the ground truth lengths. The 2D coordinates of the ends of the lines are then backprojected to the 3D world coordinates. The reprojection error is then minimized by the Estimation of Distribution Algorithm that optimizes the non-linear problem.

In Dubská et al. (2014) the authors presented one of the few fully automatic approaches and their work had a big impact on the field of automatic surveillance camera calibration. They make several assumptions such as a centre position of the camera principal point, planar ground or that vehicles move towards/from the first vanishing point. Firstly, the authors use the Hough transform mapping the 2D plane into a finite space called a diamond space by a piecewise linear mapping of lines. Through the KLT tracker linear fragment of the vehicle's trajectories are obtained and thus the first vanishing point. The second vanishing point's direction is perpendicular to the first vanishing point's direction and is also parallel to the ground. It is obtained again from the diamond space, where the edges that had already voted for the first vanishing point, are excluded. The third vanishing point and focal length are then computed. Further tangents to the silhouette of vehicles are automatically designed heading into the vanishing points and based on the statistical data of vehicle dimensions determines the dimensions of the 3D boundary box, as shown in the Figure 14. Based on the assumption that vehicles have similar widths and heights and differ in length, a scale factor is obtained this way in a fully automatic way. The authors reach the relative speed error of 2 % i.e., ± 1.5 $km \cdot h^{-1}$.

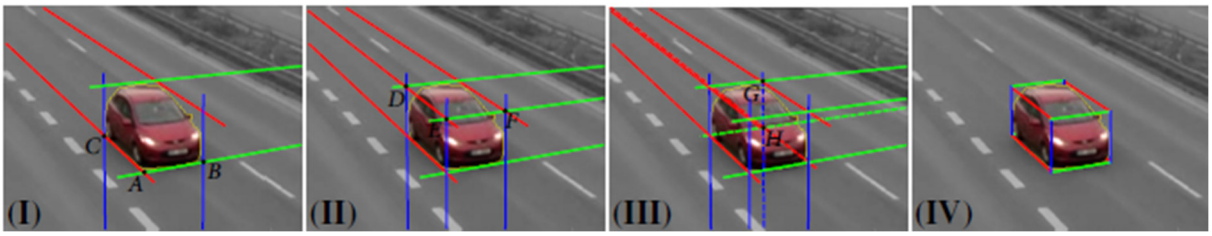


Figure 14 – 3D boundary box construction based on knowledge of three vanishing points (Dubská et al., 2014)

Similar approach presented Sochor, J. et al. (2017), where the authors identify automatically all three vanishing points in the same way as in Dubská et al. (2014), but the scale obtain by comparing dimensions of fine-grained 3D model of the corresponding vehicle class and dimensions of the detected vehicle. It is also a fully automated method that reaches a mean speed measurement error of 1.10 km·h^{-1}.

The summary of individual calibration method based on the vanishing points acquisition, their automation degree and precision, if available, is described in the Table 2.

Table 2 – Overview of Vanishing Point acquisition calibration methods, their precision and automation.

Author	Publication	Methods of Calibration based on:	Auto	Precision (if stated)
Dubská, M. et al. (2014)	Automatic Camera Calibration for Traffic Understanding.	By a Hough algorithm collect pieces of edges and by a KLT tracker vehicles' trajectories. From a diamond space obtain the first and the second VP. The third VP and the focal length is computed. Then build 3D bounding boxes and obtain a scale based on statistical data of vehicles' width and height.	✓	Relative speed error = 2 % (± 1.5 km·h^{-1})
Sochor, J. et al. (2017)	Traffic surveillance camera calibration by 3D model bounding box alignment for accurate vehicle speed measurement.	Automatic detection of vanishing points as in Dubská et al. (2014), scale obtained by comparing a 3D model of the vehicle's class with the detected vehicle.	✓	Relative speed error = 1.10 km·h^{-1}

Miles et al. (2022)	Camera-Based System for the Automatic Detection of Vehicle Axle Count and Speed Using Convolutional Neural Networks.	Project a rectangle in the road lane of known width and thus obtain the vanishing points and the scale factor.	✗	—
Zhang et al. (2021)	Vehicle localisation and deep model for automatic calibration of monocular camera in expressway scenes.	Use a deep CNN to find the vanishing points based on the parallel road markings.	✓	Straight road Mean Absolute Error of landmarks': length = 8.0 % width = 6.1 % Curved road Mean Relative Error of landmarks' length = 7.3 % Mean Absolute Error of lane's width = 3.6 %
Tang, Z. et al. (2019)	ESTHER: Joint camera self-calibration and automatic radial distortion correction from tracking of walking humans.	Known height of the camera. Connect heads and feet of at least three non-colinear positions of humans to obtain VPs.	✗	—
Wu et al. (2020)	Multi-camera traffic scene mosaic based on camera calibration.	Multiple cameras in a row, known height. The VP found as an intersection of the road lanes' lines, known lanes' width.	✗	—
Kocur and Ftáčnik (2021)	Traffic Camera Calibration via Vehicle Vanishing Point Detection.	Vehicle detection with a CenterNet, VPs then found from heatmaps using a VP detection network. The scale is not provided.	✗	The scale is not provided; thus, the errors are in the relative difference of ratios. Mean error in BrnoCompSpeed = 14.95 % Mean error in BrnoCarPark = 8.66 %

Giannakeris et al. (2018)	Speed estimation and abnormality detection from surveillance cameras.	Similar approach of finding the VPs from a diamond space as in Dubská et al. (2014). Then the third VP and focal length are computed. They refer to the original paper to match the detected vehicles to a 3D model to obtain a scale.	✓	—
Tang, X. et al. (2020)	Vehicle spatial distribution and 3D trajectory extraction algorithm in a cross-camera traffic scene.	Obtain the first VP from the intersection of the road lanes' lines. Known lane width and road markings' length.	✗	Relative distance error = 1.93 %
Zwemer et al. (2022)	3D Detection of Vehicles from 2D Images in Traffic Surveillance.	Find the first and the second VP based on the straight road lanes.	✗	—
Kocur and Ftáčnik (2020) ---------- Kocur (2019)	Detection of 3D bounding boxes of vehicles using perspective transformation for accurate speed measurement. ---------- Perspective transformation for accurate detection of 3d bounding boxes of vehicles in traffic surveillance.	Combine the method for obtaining VPs by Dubská et al. (2014) and the method of determining the scale by Sochor et al. (2017). Use the perspective transformation.	✓	Absolute speed error = 0.75 $km \cdot h^{-1}$
Liu, T. and Liu, Y. (2021)	Deformable model-based vehicle tracking and recognition using 3-D constrained multiple-Kernels and Kalman filter	Combine the PnP and the VP methods. Manually select two pairs of parallel lines that are mutually orthogonal and thus obtain the first and the second VP. Line segments' lengths are manually measured. Minimizing the reprojection error by EDA.	✗	—

The main issue in comparing individual methods is that each method is tested on different dataset under different conditions. For repeatable and representative comparing of individual methods, the recordings should be made under comparable and properly described conditions such as light conditions, camera quality, recording length, camera position, weather etc. Moreover, the precision (errors) is evaluated in various units either absolutely or relatively.

There is also a not negligible factor of various camera quality. The result in the form of precision then carries not only the information about the main calibration method, but also about the hardware quality which is the factor that can't be separated. In some methods proposed by authors also a detection phase comes prior to the calibration which brings another unwanted effect on the overall precision result.

3.4. Non-sudden braking in the traffic accidents analysis context

In a road traffic, vehicle deceleration is not only a technical parameter but also a behavioural phenomenon strongly influenced by human perception and comfort. Drivers generally avoid applying the maximum available braking capacity unless confronted with a critical situation. Instead, under everyday conditions such as approaching an intersection, a pedestrian crossing, or a red traffic light, they tend to decelerate smoothly and in a manner they perceive as comfortable. This type of braking, often referred to as *non-sudden braking*, represents a form of deceleration that does not surprise passengers, maintains vehicle stability, and remains within a range commonly practiced by most drivers in normal traffic flow.

From the perspective of forensic analysis, the concept of non-sudden braking is of particular importance. Experts frequently assess whether a driver, under given circumstances, could have avoided a collision if they had responded in a manner consistent with ordinary, non-critical behaviour. (Bradáč, A. et al., 2021). A model situation can be a traffic accident on a crossing or hitting a pedestrian or cyclist. The expert has to evaluate the technical cause of the collision between them and puts themselves questions like „How suddenly did the vehicle on a secondary road / pedestrian created a sudden obstacle to the vehicle on the main road with the right of way?" or „Could the driver on the main road have stopped before the place of collision if he/she braked only non-suddenly?". Here the non-suddenness of braking has its value because according to the Czech law Act No. 361/2000 Coll., on Road Traffic, the give a right of way means „*the obligation of a driver not to start or continue driving or a driving manoeuvre if the driver who has the right of way would have to suddenly change direction or speed.*". Following this definition, it is the key to evaluate if the driver on the main road could have stopped if he/she had braked non-suddenly.

The distinction between maximum emergency braking and comfortable everyday braking thus plays a crucial role in reconstructing the sequence of events and evaluating driver reactions in accident scenarios. However, despite its practical importance, the definition of what constitutes "non-sudden" or "comfortable" deceleration remains ambiguous. Various sources

propose different thresholds or typical values, and the lack of consensus complicates both accident reconstructions and their judicial interpretation. This gap provides space for research that aims to establish more objective and empirically based limits.

3.4.1. Review of existing approach in the Czech republic

In the Czech forensic community, several approaches are applied when estimating the threshold of non-sudden or comfortable deceleration. Since no uniform definition exists, experts rely on different assumptions and reference values, depending on their methodological preference and case context.

In the context of right-of-way accidents, Bradáč (1997) highlighted the problem of defining what constitutes a "sudden" manoeuvre. According to his interpretation, giving way means driving in such a manner that the vehicle with priority is not forced into an abrupt change of speed or direction. While the term *sudden* is not explicitly defined in legal regulations, it can be understood in practical terms. A sudden change of direction is such a deviation that it creates a risk of skid or endangers other road users, typically by forcing them into harsh braking. Likewise, a sudden reduction of speed is considered any deceleration that requires intensive braking, in contrast to merely lifting the accelerator pedal or applying mild braking.

Bradáč (1997) further emphasized that the assessment must always take into account the particular conditions of the situation, such as road surface friction. The element of suddenness lies in the fact that the manoeuvre is so unexpected that even an attentive driver could not reasonably anticipate it. In accident analysis, this implies that the expert must examine the driving behaviour of all vehicles and determine the moment when it should have become evident to the priority driver that their right of way would not be respected.

In general, there are currently three main legitimate approaches in the forensic praxis that define the non-sudden threshold, related to:

- vehicle's braking capability,
- regulatory references,
- subjective perception.

Approach related to vehicle's braking capability

One of the most frequently used approaches is to relate non-sudden braking to the vehicle's maximum technically achievable deceleration, dependent on current adhesion conditions. In this interpretation, the comfortable or non-sudden level is assumed to be approximately one half of the maximum value. For typical passenger cars, where the maximum achievable

deceleration under favourable conditions lies around 8 $m \cdot s^{-2}$, this results in a threshold close to 4 $m \cdot s^{-2}$. This value is commonly cited in expert practice as a representative limit that distinguishes ordinary braking from emergency intervention.

Approach related to regulatory references

Another line of reasoning is based on regulatory standards. Czech technical regulations, harmonized with international ECE requirements, prescribe a minimum braking performance that vehicles must achieve in order to be approved for road use. According to UN/ECE Regulation No. 13-H, which is incorporated into Czech technical regulations (Vyhláška č. 341/2014 Sb.), the minimum required mean fully developed deceleration (MFDD) for passenger cars is 5.8 $m \cdot s^{-2}$. Some forensic experts propose that non-sudden braking can be estimated as approximately half of this regulatory minimum, yielding a threshold near 2.9 $m \cdot s^{-2}$.This approach has the advantage of being directly linked to legally defined criteria, but it may be more conservative compared to the "half of maximum" assumption. (UN/ECE Regulation No. 13-H; Vyhláška č. 341/2014 Sb.)

These two perspectives illustrate the diversity of practice among Czech experts. Both rely on halving a reference value – either the maximum physical capability of the vehicle or the minimum regulatory requirement, but they lead to different numerical thresholds. The choice of approach can therefore significantly affect accident reconstruction results and the conclusions drawn about a driver's possibilities in a given situation.

Stáňa (2016) suggested in compliance with the previously written that the acceptable threshold should be considered as the smaller value of the following values:

- half of the prescribed minimum deceleration or
- half of the deceleration technically achievable given surface adhesion.

Approach related to a subjective perception

Tokař's study (2014) set out to empirically examine the boundary between "sudden" and "non-sudden" changes in speed. The motivation came from the observation that the traditional definition – a half of the prescribed minimum braking deceleration – may no longer reflect real driving conditions. To test this, a series of controlled braking experiments were conducted using several passenger cars under dry road conditions. Respondents (22 drivers and passengers of varying ages and driving experience) were asked to subjectively classify different braking manoeuvres according to four categories: safe, slightly dangerous, dangerous, and very dangerous. The experiments were performed at an initial speed of 50 kmh^{-1}, with target

decelerations of 3, 5, 7 $m \cdot s^{-2}$ and the vehicle's maximum achievable deceleration. Specialized instruments (XL Meter Pro Gamma and Pocket DAQ) were used to capture the deceleration profiles.

Considering a non-sudden change of velocity as a change that doesn't create any danger, the author assesses the threshold as a "safe" deceleration of 3.8 $m \cdot s^{-2}$. The author also considers acceptable the „slightly dangerous" deceleration of 5.9 $m \cdot s^{-2}$ coming from his experiment. As a result, these two values are combined into a value about 4.8 $m \cdot s^{-2}$. The author proposes to set the boundary not as a single value but rather as an interval in the range of 3.8-4.8 $m \cdot s^{-2}$. Moreover, provided the maximum deceleration of vehicles typically lies between 8-10 $m \cdot s^{-2}$, the range 4-5 $m \cdot s^{-2}$ should be considered.

In the following table 3, there is a summary of the currently used approaches with the thresholds:

Table 3 – Overview of current approaches to determine the non-sudden braking threshold

Approach	Threshold [$m \cdot s^{-2}$]
vehicle's braking capability	4.0
regulatory references	2.9
subjective perception	4.0-5.0

Although these approaches provide technically acceptable solutions, they also demonstrate the absence of a uniform standard. Differences in measurement methods, traffic environments, vehicle types, and driver populations lead to varying conclusions. Moreover, the boundary between "comfortable" and "emergency" braking is not sharply defined, but rather situational and context-dependent. This diversity of views underscores the need for further empirical investigation and for defining non-sudden braking in a way that is both scientifically based and practically applicable in forensic expertise.

4. PROPOSED SOLUTION OF AN AUTOMATIC TRAFFIC ANALYSIS

As it was described in the previous chapter, a direct approach of a normal driver behaviour determination was developed. It is not dependent on any artificial values from regulations nor influenced by the awareness of test drivers that are being measured. It is necessary to obtain quantities describing the comfortable driving dynamics directly by observation of a real traffic.

There are several ways to obtain such a quantities describing a natural behaviour of drivers. The first way was to manually measure desired quantities, which would be very ineffective, time consuming and inaccurate. Hence, the statistical approach of analysing a large amount of data through camera record analysis was chosen to analyse the traffic and desired quantities in the most objective, accurate and representative way.

4.1. Methodology design

In this thesis the procedure of obtaining data is generally divided into two phases: validation and data acquisition phase. The validation phase consists of a processing of simulated data under controlled conditions and parameters, setting limitations of the SW and then validating its precision directly on a particular crossing using a set of physical measurements. After ensuring that the SW can reliably process data from the particular scene, the video shots of desired manoeuvres are prepared. After an automatic processing by the tracker SW, results are evaluated statistically.

The pipeline of obtaining desired quantities is represented in the scheme in the Figure 15.

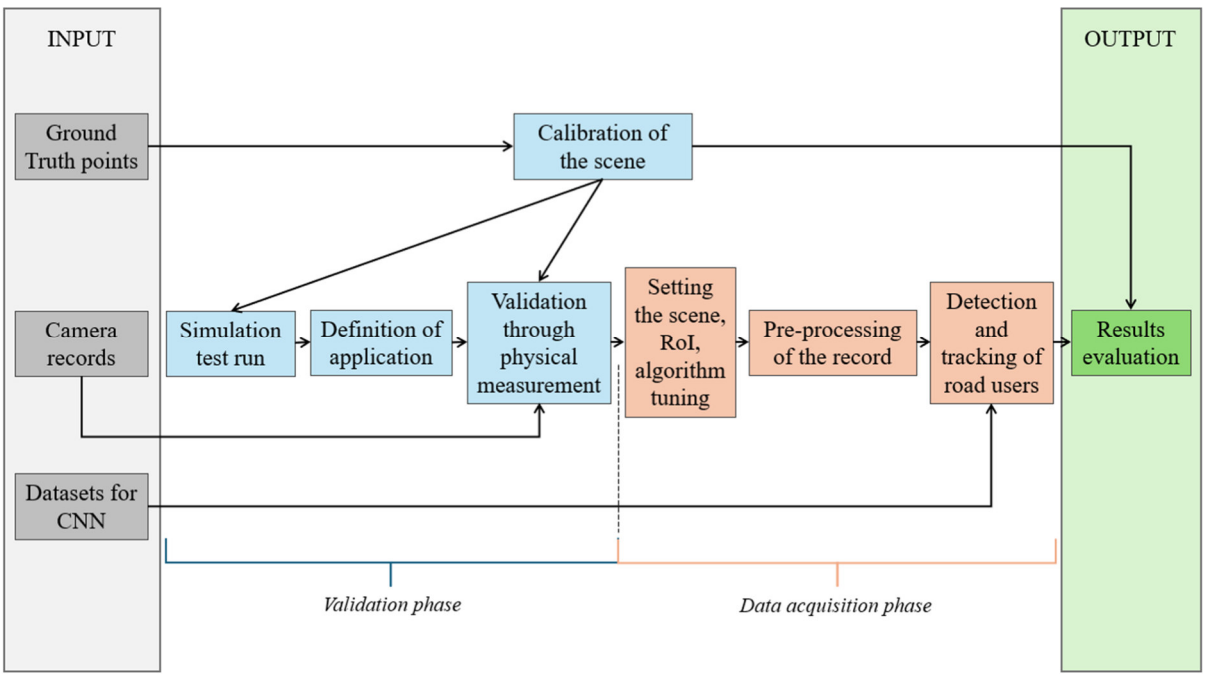


Figure 15 – Pipeline of SW validation and data acquisition (author)

As can be seen from the Figure 15, the pipeline is rather complex and build a complete solution for any traffic analysis. For the purposes of this thesis, there is also in the beginning a test run of the detection & tracking software in a simulation-based environment which helps to determine the most precise application of the SW. Then a validation on real crossing was made and after then the main data acquisition on validated scenes. The individual steps will be described in more detail further in the thesis.

Apart from a validation phase, the only manual steps include acquisition of inputs in form of camera records, ground truth points and datasets. However, ground truth points can be obtained e.g., from aerial images from maps or by knowing some standardized dimensions of particular objects in the records. Datasets are mostly already prepared and open source. Then just a little adjustment must be done to the algorithm separately for each intersection, that is supposed to be analysed, in form of determining the RoIs etc. The rest of the algorithm then runs automatically. In the following text a description of steps already made by author will be provided.

The whole implementation is being made in python, which is a high-level object-oriented programming language. The code is written in PyCharm IDE, and many in-built libraries are used for various purposes. For working with time units, a library *time* was imported, for working with mathematical expressions a library *math* and for statistical purposes libraries *statistics* and *scipy*, to be able to get parameters provided by user through command line a library *sys* has been implemented, for working with tables and saving evaluated data to standard Excel table a library *pandas* was imported. A library *json* enables to use metadata such as coordinates of selected points, ground truth points, values of pre-set parameters etc. A library *numpy* is used for working with matrixes and vectors, which is in this work very often used as the images and videos are just multi-dimensional matrixes. For working with images and video recordings, a library *OpenCV* was implemented, as it provides a broad variety of relevant functions.

4.2. Own implementation

In the following text, a model situation is described following the whole pipeline from Figure 15.

4.2.1. Input data

In the first step all input data must be prepared in sufficient amount and quality. It is naturally impossible to analyse the driving behaviour by all drivers in each situation. Therefore, a statistical sample of sufficient size will be chosen, while such sample must be representative, which means that values of desired quantities obtained in the sample must represent the

behaviour of the whole. The following assumptions were applied: no difference of normal driving dynamics between cities in Czech republic, between men and women (impossible to distinguish from the camera records), day time of recording, season, weather etc. In this model situation, only behaviour of drivers of passenger cars in a summer, daytime and dry road will be evaluated.

A big challenge of acquired video recordings is that the videos have various scales, camera angles and were recorded under various illuminating conditions. Some of the cameras also include a pollution or vibration, which are challenging factors for the detector and the tracker.

In order to determine a minimum number of samples needed for a representative statistical evaluation, a sample size of 30 was selected as it represents a widely accepted threshold in statistical analysis where, according to the Central Limit Theorem, the distribution of the sample mean approaches normal distribution. This provides sufficient reliability for statistical evaluation while maintaining feasibility in terms of data processing. A target is an evaluation of a 95% percentile assuming a normal centred distribution of a random variable, so ± 1.96 sigma interval.

Another input needed for successful working of the CNN-based detector and tracker is a dataset used for training. The YOLOv7 detector used in this work was pre-trained on the COCO dataset, a large-scale benchmark comprising over 200,000 labelled images and 80 object categories. COCO contains a wide variety of classes directly relevant to this application, including *person*, *car*, *bus*, *truck*, *bicycle*, and *motorcycle*. These categories cover the primary objects of interest in roadside traffic monitoring and provide robust representation of diverse environmental conditions, viewing angles, occlusions, and scales. Leveraging COCO-pretrained weights removes the need to collect and annotate a large custom dataset, significantly reducing development time and computational resources while maintaining high detection performance. Given the proven generalization ability of COCO-trained YOLO models in similar real-world surveillance and traffic analysis tasks, this dataset is sufficient to achieve reliable detection accuracy in the conditions of the present study. (Lin, T.-Y. et al., 2014; Bochovskiy, A. et al., 2020)

DeepSORT (Simple Online and Realtime Tracking with a Deep Association Metric) was employed to associate object detections across video frames and maintain consistent identities over time. The algorithm combines a Kalman filter for motion prediction with a deep appearance descriptor to handle object re-identification in the presence of occlusions or intersecting trajectories. In this study, the appearance descriptor was taken from the pre-trained

DeepSORT re-identification model, which has been trained on large-scale pedestrian datasets such as Market-1501, enabling robust discrimination between different individuals and vehicles based on visual features. As DeepSORT performs online learning of appearance embeddings during tracking, it adapts to the specific scene without requiring additional offline training, making it well-suited for dynamic roadside monitoring scenarios. (Wojke, N. et al., 2017; Zheng, L. et al., 2015)

The ground truth points are generally corresponding points in both image and real-world coordinates used for calibration with a PnP method. Obtaining the image-plane coordinates of points was performed using a custom script, the precise real-world coordinates of the points were acquired using a GNSS station and will be described further in a calibration chapter.

Camera records from selected traffic surveillance cameras were obtained from the City Police Department in Pardubice. The decision on which camera records will be taken for the analysis came from evaluation of the simulation test run which will be described in detail further. Basically, the result was that the SW can be reliably and with sufficient precision applied for scenes, where the camera captures at least 5 sec of the manoeuvre of interest, low occlusion and noise and highest possible resolution of the area where the manoeuvre takes place. Considering these requirements, the crossing S.K.Neumanna x Pichlova in Pardubice was chosen.

4.2.2. Pre-processing, detection and tracking

An approach for the preprocessing of the video recording and automated detection and tracking of road users will be introduced in this chapter, which contents an evolution of the solution from a background subtraction to CNN-based methods.

4.2.2.1. Background subtraction-based method

Detection & Preprocessing

To begin the process of automatic object detection, initially a classical approach based on background subtraction using a Gaussian Mixture Model (GMM) was applied. This method is widely used as a foundational step in video analysis, particularly for its simplicity, low computational cost, and suitability in controlled environments. The main idea is to model the background of a video scene using a mixture of Gaussian distributions, where each pixel is represented by a set of Gaussian functions that estimate the likelihood of it belonging to the background over time. Foreground objects are then identified as deviations from this learned background model.

The background subtraction method works on a simple principle, when it separates pixels that belong to the background from pixels belonging to the foreground. A foreground mask is obtained when the background is subtracted from an input frame. It is especially effective in scenarios where the background remains mostly stable and where quick prototyping is needed to validate the viability of automatic detection. (Berg, J. et al., 2022)

In the following sections, a description of how this method was implemented is provided together with challenges encountered during real-world application, and how the results informed the selection or adaptation of more advanced techniques in subsequent stages of the project.

The use of a simple background subtractor based on Mixture of Gaussians in OpenCV library is shown in Figure 16.

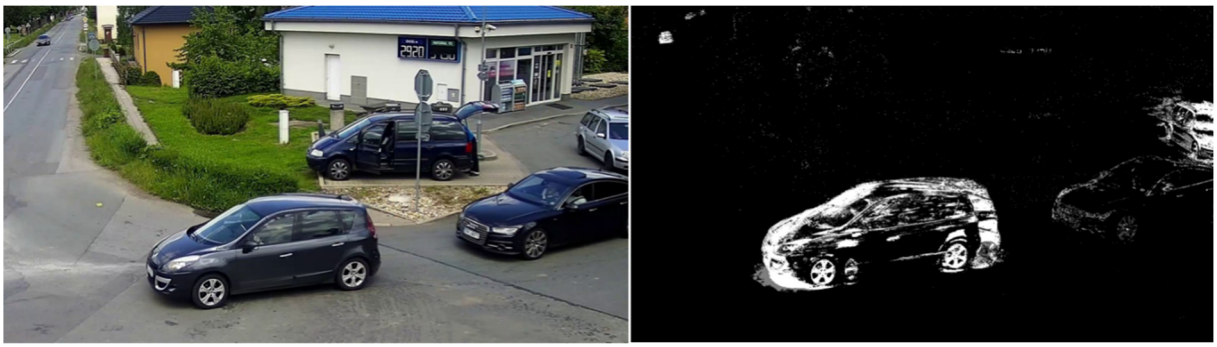


Figure 16 – Use of MOG background subtractor (author)

Nevertheless, the solution isn't that simple. As the video goes on, the program associates long-term static pixels as background and vice versa. As can be seen in Figure 16, the second vehicle in a queue slowly disappears, because it stays in one place for several seconds and the program associates its pixels as static. The first vehicle in a row accelerates, so its contours are bright, but it leaves a „ghost" behind it, because several frames ago it stayed there for a longer time. As can be seen, the parked blue vehicle with an opened trunk and driver door isn't visible at all in the foreground mask, because it stays in one place and is considered to be the background.

Moreover, the raw foreground mask suffers from noise, mostly Gaussian high-frequency noise. The raw foreground mask can be enhanced by several filters that denoise and smooth the picture. Then with the aid of mathematical morphology operations the contours of road users can be filled, and a Gaussian noise can be removed. All these operations are included in the pre-processing phase.

Initially, the detection and tracking through background subtraction has been implemented. The pre-processing phase enhances frames in order to prepare them for the detector and tracker. A compromise between the speed and the accuracy improvement must be found. In the pre-processing stage, the image from the input can be denoised by average, median or gaussian methods, that reduce high-frequency noise caused by sensor imperfections. Advanced gaussian filters can also filter out e.g., moving branches of trees or flags. Then through pre-processing methods a contrast, gamma, brightness, and histogram can be optimized, which is useful in scenes with worsened illumination like in the night, dusk or dawn. In the Figure 17 an enhanced image is displayed, where a bilateral filter was used. This filter uses multiple gaussian pixels, one for spatial blurring and the other cares for intensity difference of nearby pixels, so only those with similar intensity are blurred. This operation smoothens the image and preserves edges. On the other hand, this algorithm consumes relatively much computational power. (OpenCV – Open Source Computer Vision)

Figure 17 – Upper row: original image; Lower row: use of bilateral Gaussian filter (author)

As can be seen from the Figure 17, the image was smoothened, and the edges were maintained at the same time. However, the FPS of the original video record was 30 and by this single operation was reduced to 10 while processing.

At this point, camera calibration hasn't been implemented to the algorithm yet as the detection and tracking algorithm has been solved at first. In real, it doesn't matter the order of implementation of steps camera calibration, determining RoIs and tuning the algorithm and detection with tracking, as these processes are independent.

As the main detection algorithm was used initially background subtraction method. A function called *cv2.createBackgroundSubtractorMOG2* from the *OpenCV* library has been used. This function applies the standard background subtraction method based on Gaussian Mixture background modelling. The algorithm assigns a Gaussian distribution to each pixel's value and based on the most frequent value it estimates whether the pixel belongs to the background or foreground. The background subtractor from *OpenCV* library converts the image to the grayscale, when the lighter pixels are, the more they belong to the foreground and vice versa. The function has three main parameters that must be set:

- history,
- varThreshold,
- detectShadows.

The History parameter's format is integer and represents a number of previous frames that affect the current frame. VarThreshold stands for the threshold value (format double), that determines how old objects in the scene are still considered to be a foreground. A parameter detectShadows can remove shadows (format is Boolean) based on their shadow colour.

In the Figure 18, there is an application of this function. Parameters were set empirically as follows: history = 200, varThreshold = 100 and detectShadows = True).

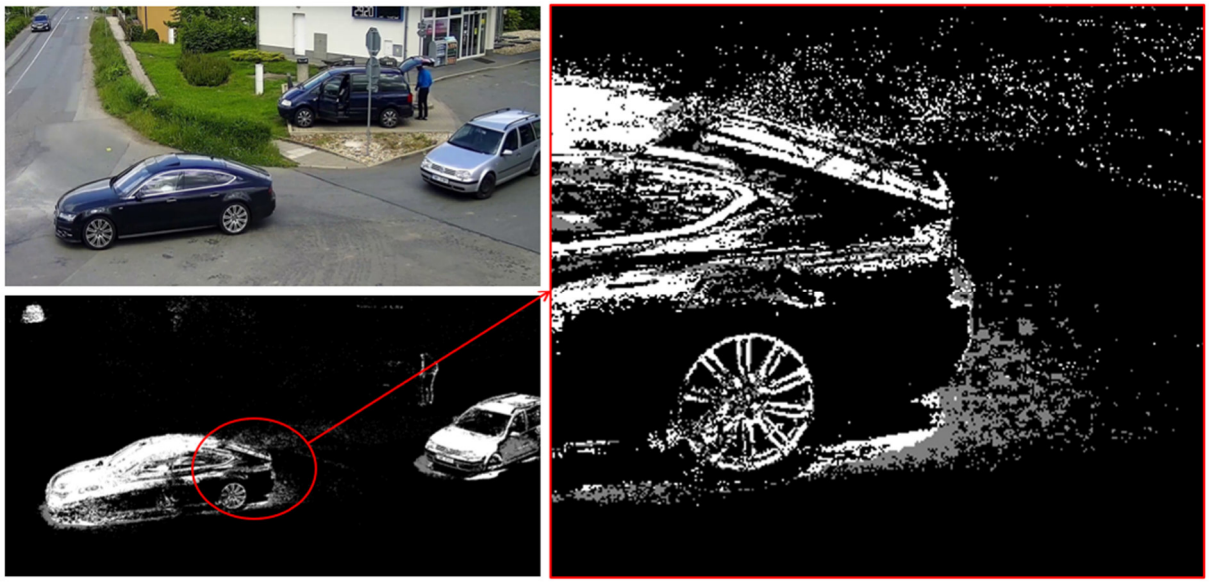


Figure 18 – Applying background subtraction method (author)

As the Figure 18 displays, moving objects like the two vehicles in the front and the one in the upper left-hand corner were detected successfully. The vehicle parked behind the two vehicles in the front isn't visible as it stays on one place for longer time and thus is considered a background. The person behind is visible because he is moving in the scene behind the trunk.

The first problem, that had to be solved, was a high-frequency noise, which is presented in the right-hand figure behind the vehicle. This problem can be effectively handled by mathematical morphology operations. Before these operations took place, it had been convenient to perform thresholding, which converts all pixels' values either to 255 (while) or to 0 (black) according to the manually set threshold. This function is a part of *OpenCV* library, and its syntax is as follows: cv2.threshold(mask, thresholdValue, maxVal, thresholdingTechnique). The first parameter is the source image, thresholdValue stands for the border, when pixels with the value higher than the border will be assigned 255 and pixels with the value equal or lower than the border will be assigned 0. The maxVal is the 255 value and thresholdingTechnique represents one of several thresholding techniques, where in this case a standard binary thresholding is sufficient.

Mathematical morphology operations

These operations are very useful when working with binary images. They can adjust contours of object, fill them, connect or separate them or reduce objects smaller than desired. There are basically two morphology operations: dilation and erosion. By dilation, the contour of object is widened, by erosion narrowed. Each of these operations is performed via a kernel, which is a matrix defined by user, that slides across the image matrix and performs multiplication between its values and overlapped values in the image matrix.

By combining these two basic operations, a morphological opening and morphological closing operations are obtained. Morphological opening is a mathematical operation, where an erosion followed by a dilation take place. It is useful when a reduction of high-frequency noise or small unwanted objects is desired. Firstly, by erosion these unwanted small objects are erased and then by dilation the original thickness of big objects is restored. In the Figure 19 a reduction of high-frequency noise in displayed, which was performed by opening function with kernel created of matrix 2x2 filled with ones.

Figure 19 – Performing morphological opening in the right-hand figure (author)

As it can be seen, the high frequency noise has been successfully removed. In the next stage it is necessary to fill the contour of the vehicle by the operation of morphological closing, which is performed in the Figure 20.

Conversely to the morphological opening, in morphological closing the dilation is used at first, which can fill smaller holes, bays or gaps. Then the morphological erosion process thickens the overall shape of the objects to the former thickness.

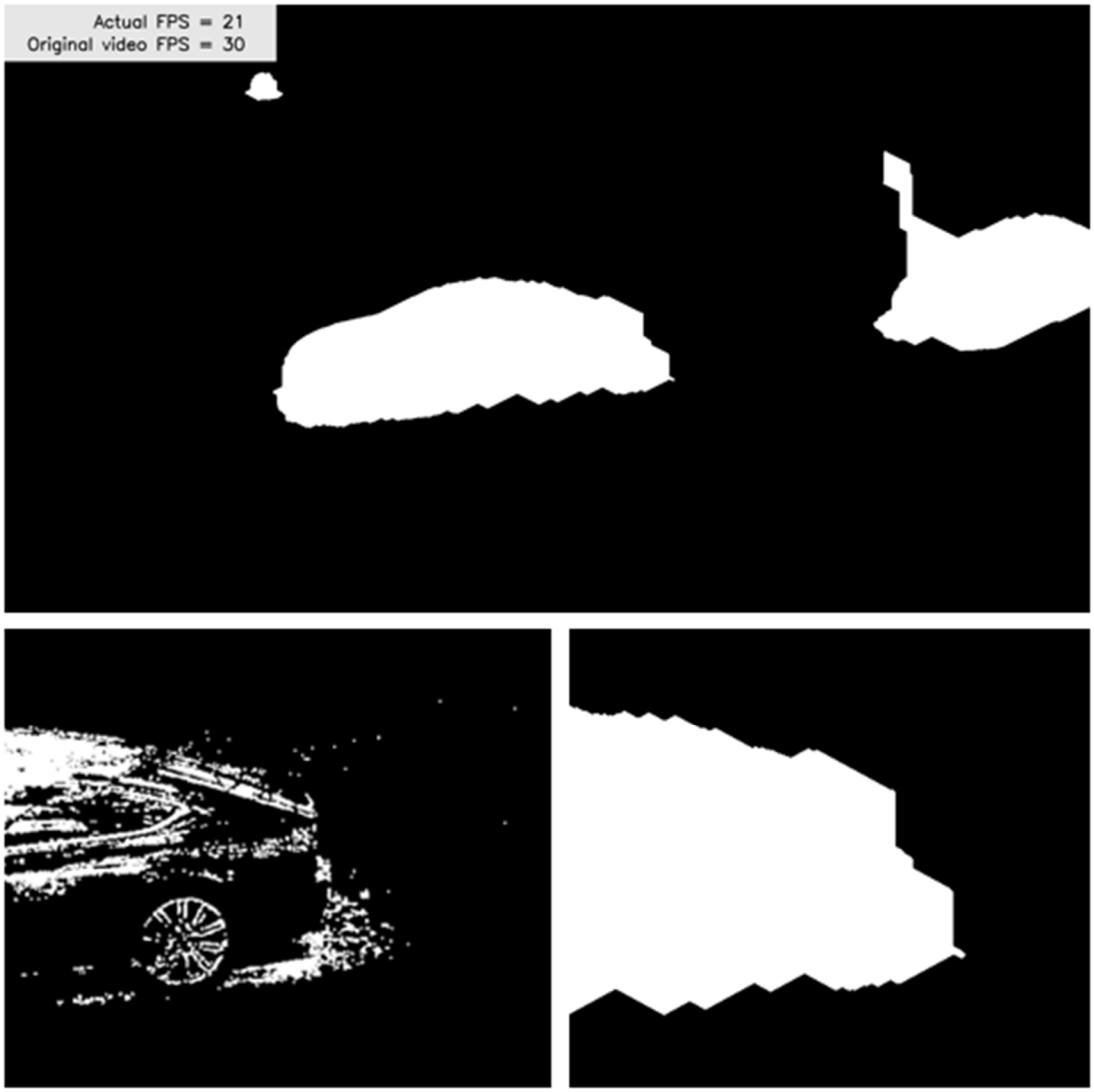


Figure 20 – Application of the morphological closing (author)

At this point all moving objects in the scene form closed shape that is similar to their outer contour. To obtain such shapes, an opening operation had to be done twice with a 3x3 cross kernel, which was designed empirically. The subsequent closing operation had to be performed 40x with a larger kernel 5x5 forming an ellipse.

$$opening\ kernel = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \qquad closing\ kernel = \begin{bmatrix} 0 & 0 & 3 & 0 & 0 \\ 3 & 3 & 5 & 3 & 3 \\ 3 & 5 & 7 & 5 & 3 \\ 3 & 3 & 5 & 3 & 3 \\ 0 & 0 & 3 & 0 & 0 \end{bmatrix}$$

By performing the background subtraction, binary thresholding, morphological opening and closing multiple times, the speed of the algorithm reduced from original 30 FPS to 21 FPS (the bilateral filtering must had been removed).

At this point, a problem appeared that by setting kernels and other parameters to perfectly detect shapes of large objects (objects close to the camera), the smaller objects in the distance

disappeared due to the opening operation. Conversely, when the parameters were set to detect properly the small objects in the distance, the large objects were divided into several smaller objects due to the closing operation.

This problem solved a newly proposed method in this thesis, the adaptive masking. The algorithm firstly detects large object and assigns them an ID. Then the parameters are automatically set to perfectly detect smaller objects, ignoring the areas, where larger objects were detected. By concatenating these two sets of objects a final mask with all objects is obtained.

However, when more objects are moving partially occluded e.g., vehicles driving towards or from a camera in a row, the detection with the background subtraction mostly fail and assign the blob of more objects with only one ID thinking that it is a one big object.

Tracking

Multiple object tracking (MOT) is the main bottleneck of the background subtraction method. When having a record with just few moving objects, that are well visible and don't stop, this method can be used with relatively high precision. But most of real situations aren't so ideal and the tracking algorithm must handle issues like occlusion of objects, keeping the object although it stopped etc. In the following chapter a brief explanation of problems encouraged by implementing the background subtraction method for tracking will be provided.

As mentioned before and as proceeds from the principle of background subtraction method, when an object stops in the image, its pixels stop to change their values. Their new values start to accumulate in the Gaussian mixture model for each of them and, depending on the „*history*" parameter value in the background subtraction method, they after several seconds change to the background. Then the program doesn't see the staying objects anymore, can't evaluate their interaction with others and when they accelerate again, the program gives them a new ID thinking, that these objects are new ones. These situations happen in intersections often when the vehicles stop in front of the traffic red light, pedestrian crossing or due to traffic congestion. Pedestrians stop in front of pedestrian crossings waiting for cars to stop etc. This problem was solved by firstly assigning each object as a so called „*blob*" with unique ID and calculating its velocity, acceleration and direction vector. When the object's velocity over several frames kept decreasing, its deceleration was negative, centre wasn't close to the edge of the image and simultaneously the object's direction vector wasn't facing out of the image, when such object stopped and disappeared, the algorithm assigned its state as „*waiting*" and let its ID active. Then when the object started to move again and appeared, a condition, that a newly detected object

close (to some defined radius) to the centre of the waiting object is the old waiting object, assigned the old ID to this newly moving object.

The main problem, that couldn't be overcome in the background subtraction method, was the occlusion. Occlusion of objects is a common phenomenon; objects can be partially or fully hidden behind each other or behind solid objects in the scene. When the tracked object disappears in the middle of the image, the tracking algorithms make prediction depending on its kinematic quantities before where the object should emerge after some frames or seconds. However, this algorithm doesn't consider the appearance of the object, so if in the middle of the image appear more objects close to each other, the re-ID process usually fails.

The main advantages and disadvantages of this method can be summarized as:

PROS:	● Robust by illumination changes. ● Shape precision by individually moving objects. ● Easier tuning of the algorithm. ● No need of object dataset.
CONS:	● High impact on computing speed. ● Severe occlusion problems. ● Problems with non-moving objects. ● Classification of objects isn't included in this method. ● Objects can break up to more pieces or conversely more objects can be considered a one big object. ● Complicated system of conditions when dealing with all problems.

For these reasons the method of background subtraction was refused to be used in this program and other more robust method will be proposed to better suit the requirements of this thesis.

4.2.2.2. CNN-based methods

Methods of automatic object detection and tracking based on CNNs are state-of-the-art in the field of computer vision and machine learning. The next step after concluding, that the

background subtraction method isn't suitable for this application, was to choose the most suitable method based on deep learning to be implemented to the SW.

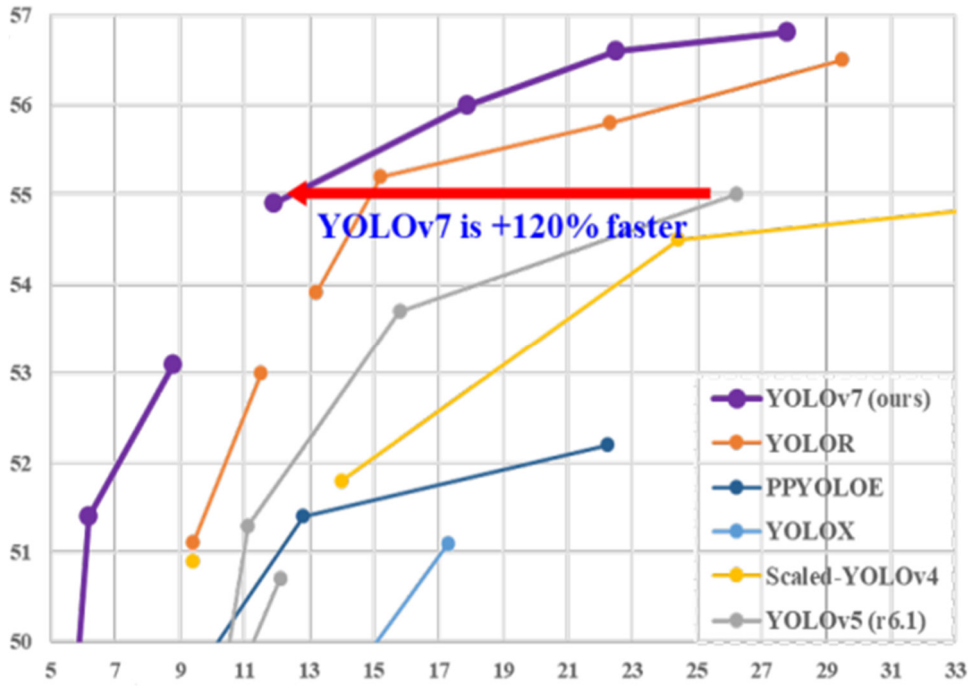
A brief overview of such methods was provided in the chapter 3.1.2. Finally, an automatic detection method YOLOv7 has been selected so far to detect desired objects. The main reasons to choose this method were its speed, further implementation abilities and its accuracy. YOLO algorithms are known to be one of the fastest and are able to perform reliable real-time detections. There are also many modifications of YOLO improving its speed and accuracy. This method will be complemented by DeepSORT tracker, which were chosen due to its ability to handle occlusion well, which is the main problem in tracking objects in traffic scenarios. Moreover, DeepSORT uses its own CNN to work with appearance features of objects which makes it more successful when dealing with occlusion.

YOLOv7 follows the "single-shot" approach to object detection, where an image is processed in a single pass by a neural network that directly outputs both the locations (bounding boxes) and categories of detected objects (Wang et al., 2022). The network consists of three main parts: a backbone (for extracting image features), a neck (for combining features from different layers), and a head (for predicting bounding boxes and classes). YOLOv7's backbone is based on the Extended Efficient Layer Aggregation Network (E-ELAN), which uses advanced feature aggregation to maintain strong information flow and stable training even in very deep models (Ali et al., 2024). This enables the system to process images rapidly while still recognizing objects with high precision. Unlike some earlier models, YOLOv7 is trained directly on the target detection dataset (MS COCO) without the need for pretraining on classification datasets, simplifying the pipeline while maintaining high accuracy (Ali et al., 2024; Wang et al., 2022).

YOLOv7 introduces a set of enhancements, described as a trainable "bag-of-freebies", which improve detection performance without increasing the computational cost during inference (Wang et al., 2022). These include planned re-parameterized convolutions (RepConvN), which restructure convolutional layers for more efficient computation, and a coarse-to-fine auxiliary head that provides extra training guidance but is removed at inference time (Ali et al., 2024). Other improvements include integrating batch normalization into convolution weights, applying implicit knowledge transfer techniques from earlier YOLO versions, and using an exponential moving average of model weights to improve stability (Ali et al., 2024; Wang et al., 2022). Compared to predecessors like YOLOv4 and YOLOR, YOLOv7 reduces parameters by up to 75 % and computation by 36 %, while improving

detection precision by about 1.5%, achieving state-of-the-art real-time performance (Ali et al., 2024; Wang et al., 2022).

The v7 version was developed in 2022 and has many improvements against the older YOLO versions like the use of a better back-bone architecture E-ELAN, that makes faster the process of machine learning and training and others. In the official paper introducing this method (Wang et al., 2022), the comparison with other selected state-of-the-art methods has been provided (Graph 1). On the horizontal axis a time [ms] can be found and on a vertical axis mean Average Precision (mAP) [%] measured using the same dataset MS COCO as the currently proposed SW in this thesis.



Graph 1 – Comparison of the selected YOLO state-of-the-art detection methods (Wang et al., 2022)

A Figure 21 shows the object detection with the YOLOv7 algorithm.

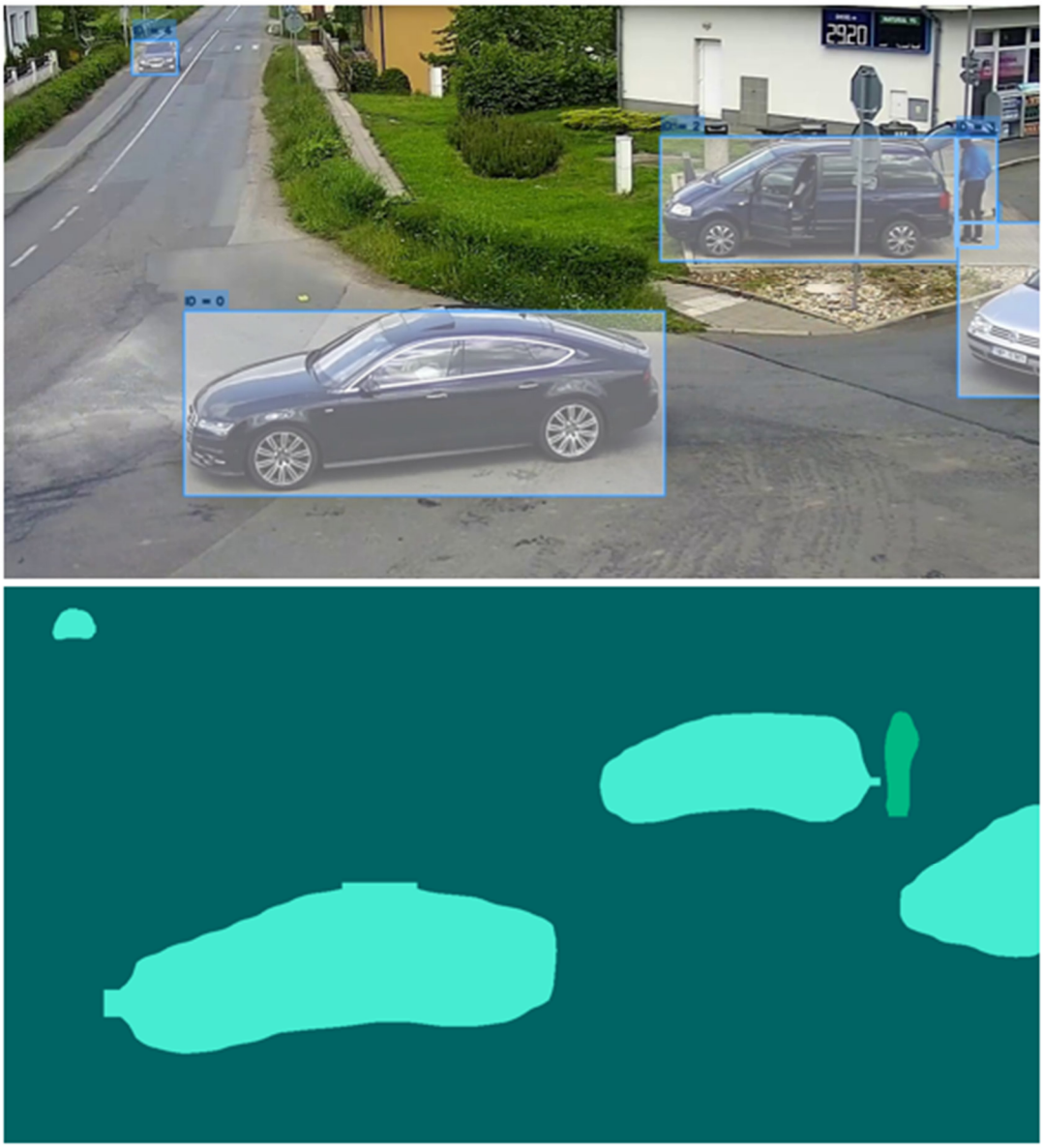


Figure 21 – Detecting objects with YOLOv7 (author)

And also, a detection in a traffic with a big number of vehicles, when some of them are also occluded (Figure 22).

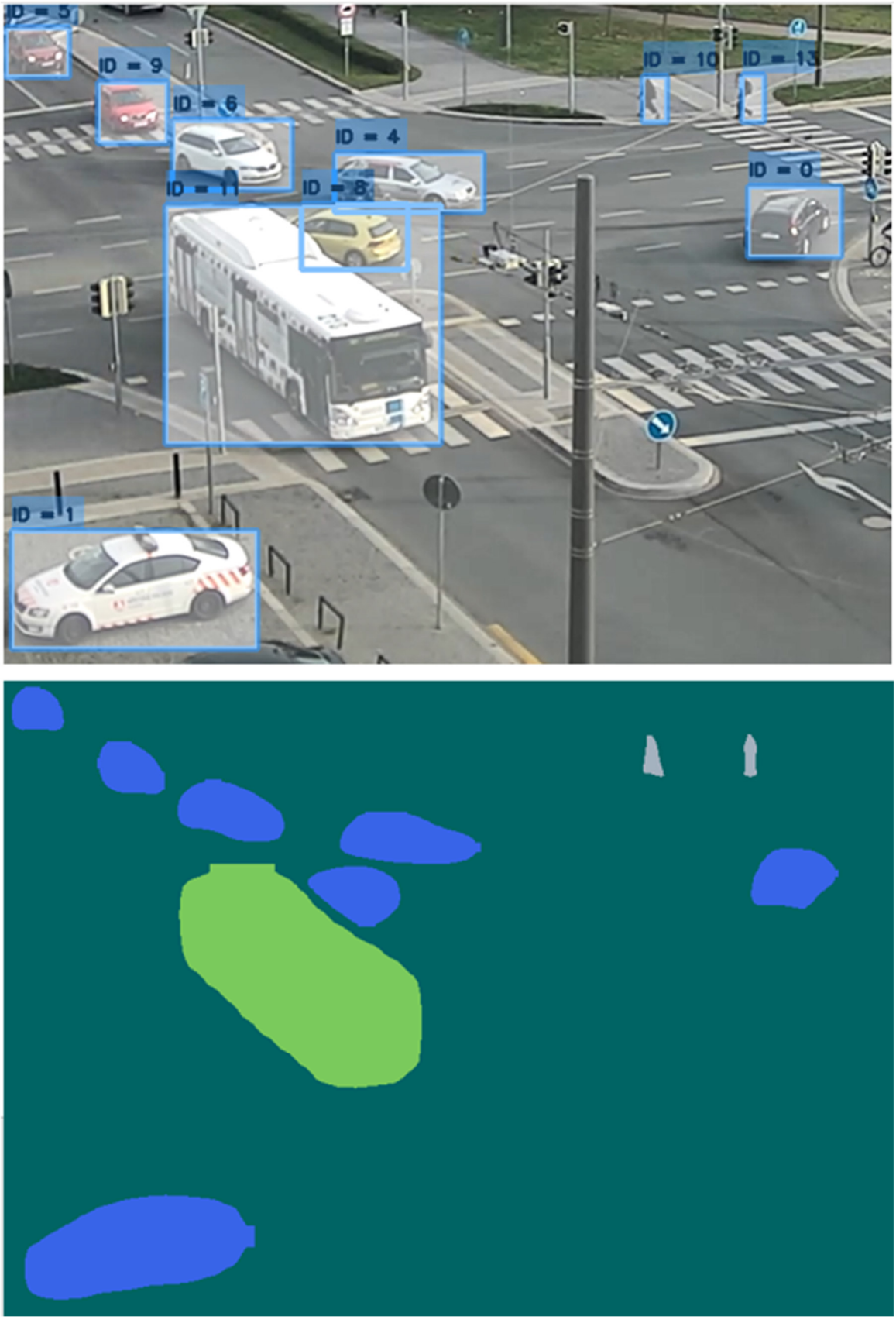


Figure 22 – Detecting objects with YOLOv7 (author)

As can be seen from the Figures 21 and 22, YOLOv7 is a powerful tool for precise and fast detection of road users. It can successfully deal with occlusion, segmentation, worse resolution of the video and various scales of objects. Moreover, in the lower figure there is a segmentation figure, where individual classes of objects are displayed by different colours. So, personal vehicles are in blue colour, bus in green and pedestrians in grey.

Tracker

After a successful detection, localization and classification of objects, a DeepSORT tracker is implemented to track the path of the objects. DeepSORT (Deep Simple Online and Realtime Tracking) enhances the original SORT framework by integrating long-term appearance cues into the typical short-term, motion-based tracking. Each detected object is tracked using a Kalman filter to predict its next position, while a CNN trained for person/object re-identification extracts an appearance embedding. These two cues, motion and appearance, are combined into a single cost metric optimized by the Hungarian algorithm to associate detections with existing tracks. This design significantly improves identity persistence during occlusion and reduces false matches while preserving real-time processing speeds (Wojke, N. et al., 2017).

Relative to traditional motion-only trackers like SORT (Bewley et al., 2016), DeepSORT markedly decreases identity switches by about 45 % on benchmark datasets thanks to the appearance descriptor that robustly differentiates similar-looking objects even when their trajectories overlap or become obscured (Wojke, N. et al., 2017; Lin, S. et al., 2021). This makes DeepSORT especially well-suited for traffic surveillance and vehicle tracking applications (e.g. using YOLOv7 + DeepSORT) where occlusions, re-identifications, and high throughput are commonplace.

In a following Figure 23, a camera record of one of the test drives is displayed. The car is successfully detected and tracked with an ID = 8. Then the car disappears below the tree for ca. 3 seconds and appears ca. 15 m further. As the tracker learned the appearance and motion of the car, it performed the re-ID process successfully back to ID = 8.

Figure 23 – Object occlusion handling by DeepSORT tracker (author)

4.2.3. Simulation test run

After obtaining and configuring the YOLOv7 detector and implementation of the DeepSORT tracker, it was advisable to perform a test run of the SW to verify its functionalities

and identify weaknesses. For this purpose, a simulation in a SW for traffic accidents analysis, Virtual Crash 3, has been prepared. A simulation provides an isolated environment with no occlusions, noise etc. which can be fully customized by the user, e.g. camera position, vehicle path relatively to the camera, kinematic and dynamic parameters are defined and known and also calibration points (ground truth real-world points).

In order to be able to identify the precision, strengths and weaknesses and functionality of the program under various controlled circumstances, the simulations were performed with one car (Škoda Octavia) driving on a straight plain road with two variables: initial speed & deceleration and camera angle towards the longitudinal axis of the road. These configurations were varied as follows:

Initial speed & deceleration

a) $v_{konst} = 50\ km \cdot h^{-1}$,
b) $v_{init} = 50\ km \cdot h^{-1}$ and deceleration $4\ m \cdot s^{-2}$,
c) $v_{init} = 80\ km \cdot h^{-1}$ and deceleration $7.649\ m \cdot s^{-2}$*.

*for the chosen Škoda Octavia car this was a limit in the Virtual Crash 3 SW.

Camera angle with road

I. 0° (camera and road longitudinal axes parallel),
II. 30°,
III. 60°,
IV. 90° (camera and road longitudinal axes perpendicular).

Combining all these variable levels, 3x4 analyses were performed and evaluated. In the following text only b-III particular example will be shown ($v_{init} = 50\ km \cdot h^{-1}$ and deceleration $4\ m \cdot s^{-2}$ and camera angle 60°).

The following Figure 24 displays the test scene.

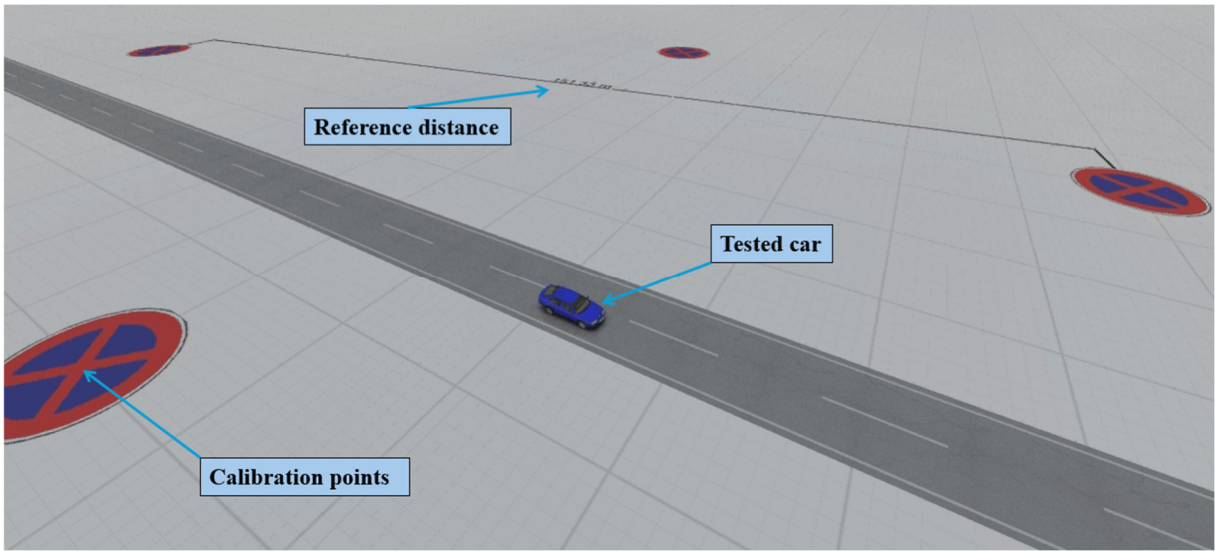


Figure 24 – Test run in a simulation environment, camera angle 60° (author)

In each scene, four calibration points were placed. Their mutual distance was known and was used as a ground truth for a calibration of the scene. The highest distance is visible in the Figure and is used to define a scale for the calibration.

It was proven by the testing that the calibration points layout is especially important and has a direct influence on a calibration precision. By placing the four calibration points, the road plane is defined, which implies the main rules for defining these points: the points must not be colinear and shall be placed across the whole plane (not only along the road). The following Figure 25 shows the difference between the correct and incorrect calibration points layout.

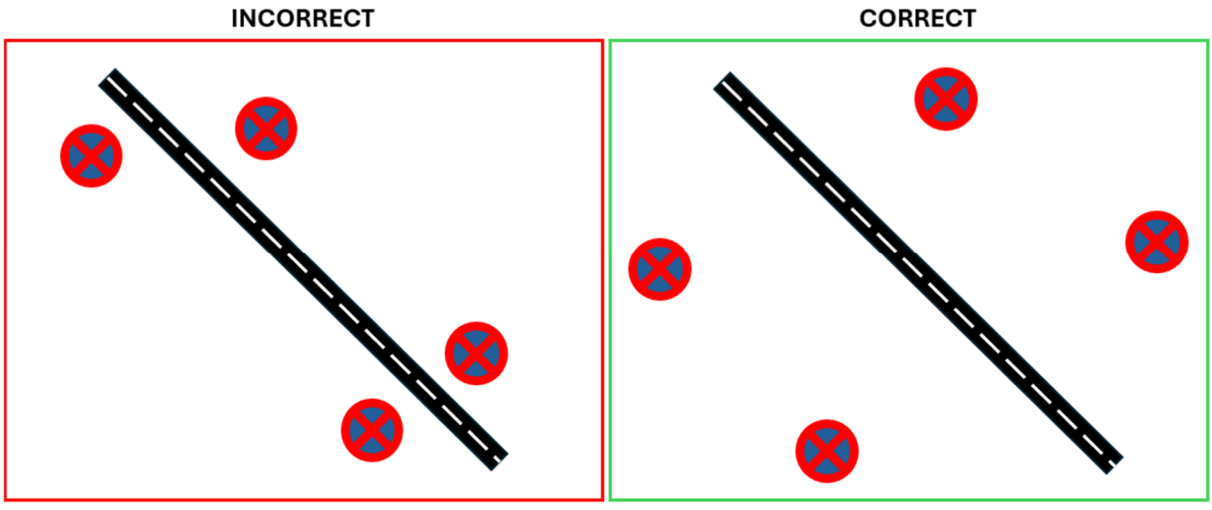


Figure 25 – Layout of the calibration points (author)

For the calibration, two ways of obtaining the homography matrix were tried: first by following the equations from the chapter 3.3.1 and performing the non-linear optimization, the

other by using a built-in function cv.findHomography. The cv.findHomography function provided always the most accurate homography matrix. For the particular case b-III, the calibration result was as follows:

$$H = \begin{bmatrix} 0.3492052 & 1.73713063 & -119.63791686 \\ -0.04197064 & 1.558077 & -68.04181827 \\ -0.00002131 & 0.01338669 & 1 \end{bmatrix} \quad (24)$$

The comparison between ground truth points and projected image-plane points using the homography matrix is provided below. Each matrix consists of 4 calibration points (rows) having x and y coordinate (columns) in metres.

$$ground\ truth = \begin{bmatrix} 95.09 & 86.56 \\ 28.39 & 13.96 \\ 130.77 & 4.19 \\ 173.9 & 55.53 \end{bmatrix} \quad (25)$$

$$projected\ points = \begin{bmatrix} 95.08999634 & 86.55999756 \\ 28.38999939 & 13.96000004 \\ 130.77000427 & 4.19000006 \\ 173.8999939 & 55.52999878 \end{bmatrix} \quad (26)$$

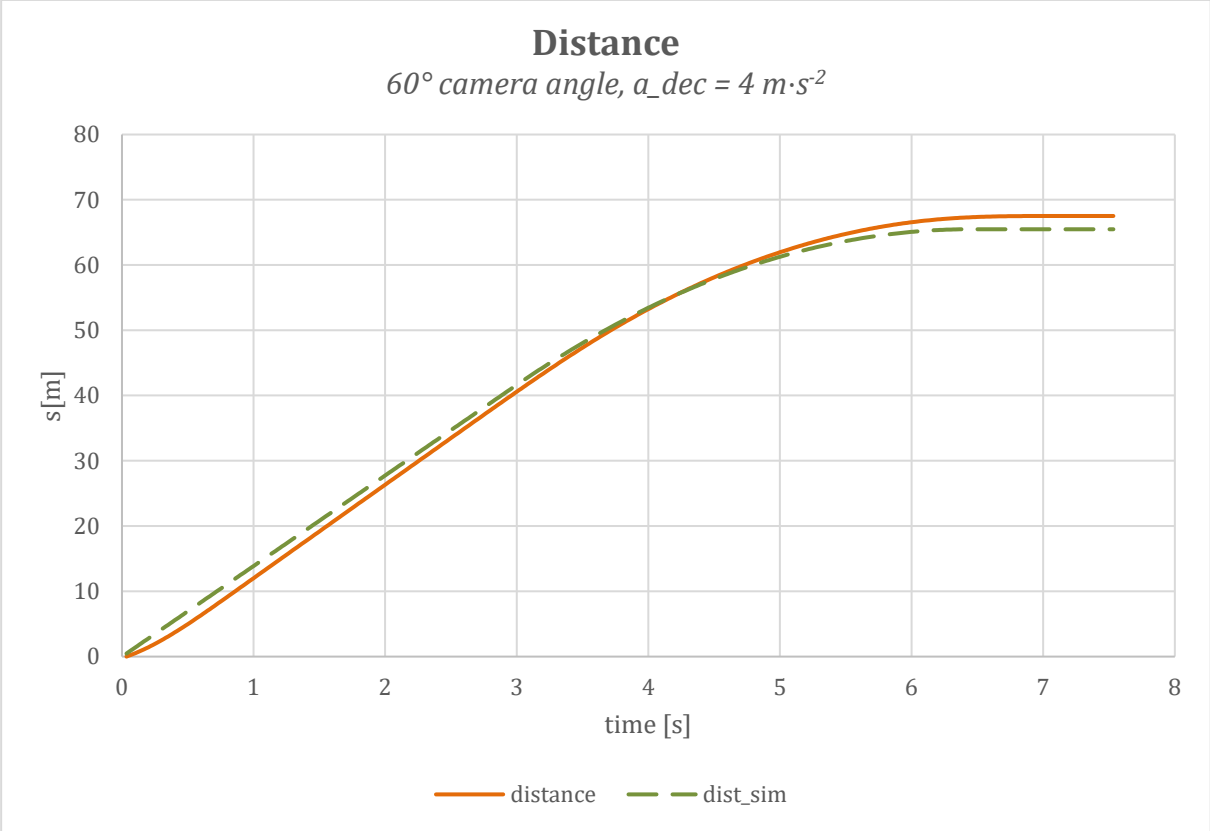
Hence the difference matrix:

$$Differences = \begin{bmatrix} -0.00000366 & -0.00000244 \\ -0.00000061 & 0.00000004 \\ 0.00000427 & 0.00000006 \\ -0.0000061 & -0.00000122 \end{bmatrix} \quad (27)$$

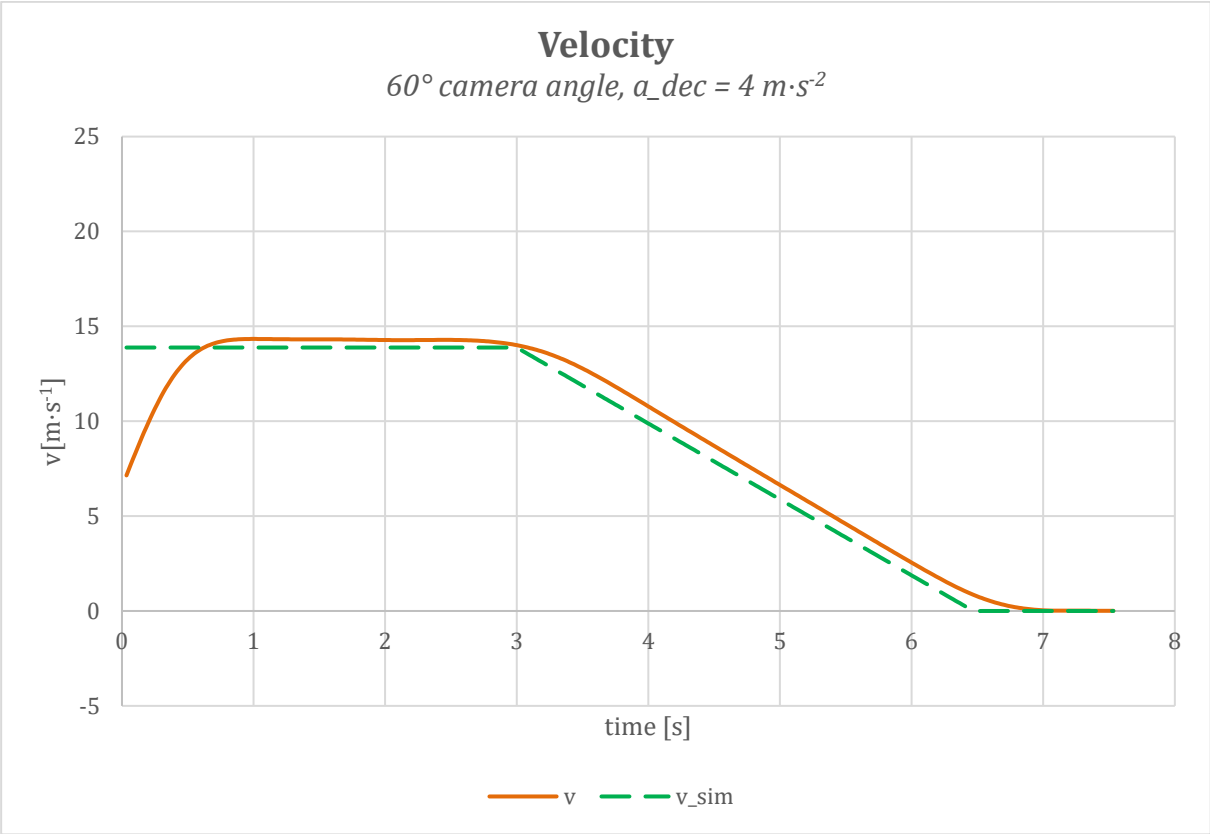
As can be seen, the calibration was successful providing a sufficient precision and robustness across the whole plane. The differences are negligible. For sure, the calibration is partially manual as the user has to mark the centres of the calibration points which includes a small error of the locating the cursor at the centre of the points and clicking. The error can be worse at the calibration points lying in a perspective, however, for a practical use of this approach is also negligible.

For the particular case, the vehicle started moving 50 km·h^{-1} for the first 3 seconds and then started to decelerate by 4 m·s^{-2}. After capturing the timestamp (frame number + FPS), and x and y coordinate of the bounding box of the car, the data were downloaded from the database and needed to be smoothened. The smoothing procedure is described in detail in the chapter 3.2.7. Data acquisition and processing.

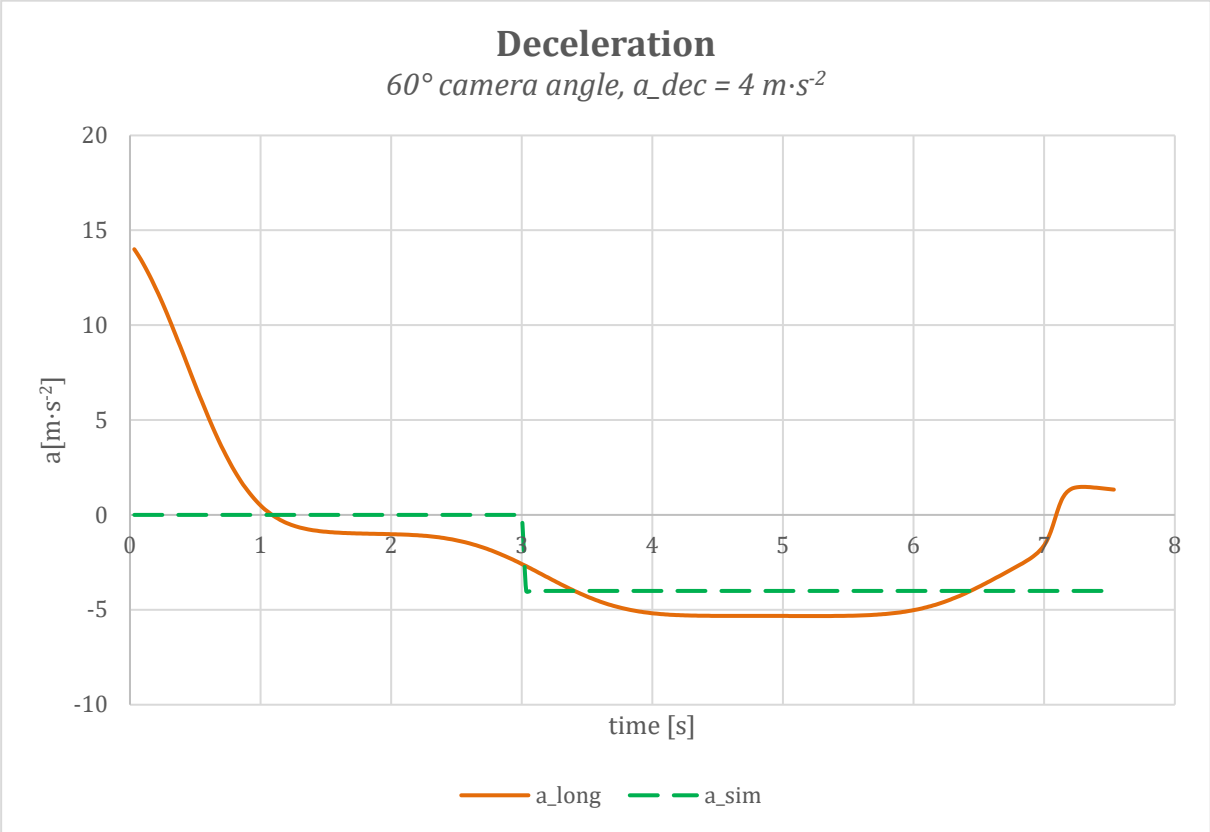
The graphs 2-4 comparing the ground truth quantities to quantities evaluated and computed from the detection & tracking SW are provided below (distance, velocity in the longitudinal axis of the car, acceleration in the longitudinal axis of the car):



Graph 2 – Distance evaluation from simulation (50 km·hr^{-1}, -4 m·s^{-2}, 60deg).



Graph 3 – Velocity evaluation from simulation (50 km·hr^{-1}, -4 m·s^{-2}, 60deg).



Graph 4 – Velocity evaluation from simulation (50 km·hr^{-1}, -4 m·s^{-2}, 60deg).

The graphs look remarkably similar for all the configurations of speed and deceleration (v_{konst} = 50 km·h^{-1}, v_{init} = 50 km·h^{-1}and 4 m·s^{-2} deceleration, v_{init} = 80 km·h^{-1} and 7.649 m·s^{-2} deceleration) and for all the camera angles 30°, 60° and 90°. Only for camera angle 0°, when the longitudinal axis of the camera is parallel to the longitudinal axis of the road, the results are imprecise due to a high impact of the perspective distortion.

As can be seen from the Graph 2 (Distance), the two curves are remarkably close to each other. It is mainly due to a good precision of the calibration, detection and tracking. The core data that are obtained from the SW are coordinates which are directly converted to the distance, so there is no additional error multiplication based on a mathematical conversion. The velocity curves are also close to each other, there is only a small offset (~ 0.4 m·s^{-1}) visible which can be caused by slightly not matching the centre point of the bounding box and the mass centre which is measured in the simulation SW. The other cause could be also a little mismatch in the timestamp (the time is calculated from FPS and frame number and each frame is passed to detector, then to the tracker etc. which can cause small delays). The small error on the distance level is amplified by the deriving the distance function to obtain the velocity. Considering the acceleration curves, there is a visible delay between the „perfect“ curve from the simulation and the curve evaluated from the tracked data. After the application of the deceleration at 3 seconds, the evaluated curve „reacts“ with a ca. 1 second delay and is offset by ca. 1.3 m·s^{-2}, where the cause is the same as per velocity curve. Moreover, after two deriving of the distance function, a small discrepancy on the distance level increases in the acceleration level.

In the following section, the differences between the ideal curves (from simulations) and evaluated curved (from the SW) will be provided. The difference will be evaluated only for selected sections of interest in the graph, mainly after ca. 1 second after a transitional event like change of acceleration value or after the initialization of the SW. This way the evaluation of the difference won’t be distorted by transitional effects and will represent the process of obtaining desired quantities.

To quantify the agreement between the simulation-based and detection-based velocity profiles, two complementary metrics were used: Root Mean Square Error (RMSE) and Normalized RMSE (NRMSE). The RMSE measures the average magnitude of deviations between two datasets in the same units as the measured variable:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(v_{SW} - v_{sim})^2} \tag{28}$$

where v_{sim} is the simulated value and v_{SW} the detected value at time step i. Lower RMSE indicates closer agreement, with values less than 5-10 % of the variable's range generally considered acceptable.

The NRMSE removes the influence of measurement scale by normalizing RMSE with the range of the reference dataset:

$$NRMSE = \frac{RMSE}{\max(v) - \min(v)} \tag{29}$$

NRMSE below 0.05-0.10 denotes acceptable accuracy; higher values indicate excessive deviation.

The following table displays RMSE and NRMSE for each configuration of the simulation for the distance, velocity and acceleration.

Table 4 – Difference between ideal (simulated) and SW-obtained evaluated curve.

Configuration		RMSE	NRMSE
0°, v_{konst} = 50 km·h^{-1}	Distance	0.45	0.04
	Velocity	1.21	N/A
	Acceleration	2.66	N/A
0°, v_{init} = 50 km·h^{-1}, -4 m·s^{-2}	Distance	0.51	0.05
	Velocity	1.57	0.09
	Acceleration	1.88	0.12
0°, v_{init} = 80 km·h^{-1}, -7.649 m·s^{-2}	Distance	3.12	0.08
	Velocity	2.80	0.11
	Acceleration	4.41	0.14
30°, v_{konst} = 50 km·h^{-1}	Distance	0.52	0.03
	Velocity	0.8	N/A
	Acceleration	0.14	N/A
30°, v_{init} = 50 km·h^{-1}, -4 m·s^{-2}	Distance	1.25	0.05
	Velocity	0.33	0.07
	Acceleration	0.65	0.08
30°, v_{init} = 80 km·h^{-1}, -7.649 m·s^{-2}	Distance	2.11	0.06
	Velocity	0.79	0.06
	Acceleration	0.34	0.09
60°, v_{konst} = 50 km·h^{-1}	Distance	0.85	0.03
	Velocity	0.41	N/A
	Acceleration	1.00	N/A
60°, v_{init} = 50 km·h^{-1}, -4 m·s^{-2}	Distance	0.58	0.04
	Velocity	0.62	0.06
	Acceleration	2.27	0.07
60°, v_{init} = 80 km·h^{-1}, -7.649 m·s^{-2}	Distance	1.10	0.06
	Velocity	0.45	0.08
	Acceleration	0.87	0.10
90°, v_{konst} = 50 km·h^{-1}	Distance	0.31	N/A
	Velocity	0.43	N/A
	Acceleration	0.59	0.09

90°, v_{init} = 50 km·h^{-1}, -4 m·s^{-2}	Distance	0.47	0.04
	Velocity	0.52	0.07
	Acceleration	0.72	0.11
90°, v_{init} = 80 km·h^{-1}, -7.649 m·s^{-2}	Distance	1.89	0.07
	Velocity	0.44	0.09
	Acceleration	0.73	0.08

As can be seen from the Table 4, generally the higher values of RMSE are located in the 0° camera angle, which can be technically explained by higher perspective distortion and lower resolution of the movement of the vehicle. On the other hand, most of the values are <1 which means that the mean error or distance, velocity and acceleration is acceptable. The same situation is at the NRMSE which characterizes the value of the RMSE relative to the entire range of the function. Also, here the result is acceptable as mostly the NRMSE is below 10 %. In case there is N/A value in the NRMSE it means that the driving function is constant (v_{konst}, then also acceleration is constantly zero) and NRMSE cannot be evaluated. Sometimes outliers appear in the Table 4 and this is one of the reasons why also a physical validation of the SW is necessary.

The main goal of the simulation test run wasn't to have the simulated and SW evaluated curves 100% equal but to identify the way the SW works in a most robust way and situations that the user must avoid. Below the key learned points from the simulation test run are listed:

- correct layout of the calibration points must be determined,
- camera angle to be ≥ 30° (<30° acceptable with additional validation or when the camera is high enough to obtain a sufficient resolution of the movement),
- delay of evaluated deceleration curve is 1 sec.

4.2.4. Validation through physical measurement

In order to provide precise validated measurement of the real traffic and drivers' behaviour, it was necessary to perform a validation measurement on the real crossing. Considering the recommendations for calibration coming from the simulation test run, a crossing S.K.Neumanna x Pichlova in Pardubice was chosen. The camera is located high above the road which provides sufficient resolution of the movement of measured objects. The camera is located on a roof of a surrounding building which is higher than on most of the other crossings where the cameras are mostly attached to the streetlamp pole. The video recordings were obtained in the cooperation with the City Police Department of Pardubice. The view of the cameras is displayed in the following Figure 26:

Figure 26 – Crossing chosen for analysis (S.K.Neumanna x Pichlova) (author)

The surveillance cameras are static on this crossing which enabled to measure firstly kinematic and dynamic quantities with a test vehicle and after it evaluate the other drivers at any time.

4.2.4.1. Calibration

There are various methods to be used to obtain ground truth points and distances in the image. It is obvious in many relevant papers, that authors always look for the most convenient compromise between a degree of calibration automation and accuracy. In this thesis, the number of analysed intersections is not so high to make a short manual calibration ineffective. Moreover, each intersection is different, and it is necessary in this case to define RoIs manually to obtain data in the best possible quality from the desired areas.

For this reason, a semi-automatic calibration method was used, when ground truth points were obtained from a physical measurement in the crossings using a Leica station which was provided by the Faculty of Transportation Sciences CTU in Prague, Department of Forensic Experts in Transportation. For the measurement, a combination of Leica CS20 controller and Leica GS18 GNSS RTK Rover smart antenna was used. The controller supports raw GNSS data logging with frequencies up to 20 Hz which enabled both the static and dynamic measurements. The GS18 antenna was firstly mounted to the pole for a static measurement of calibration points with a precision of 1-2 cm. For the static measurements, the controller enables user to switch on a tilt compensation and visualize the measured points directly on a map. After capturing the static calibration points, the GS18 antenna was switched to a dynamic measurement mode, when the tilt compensation was turned off, the antenna was attached on the roof of the test vehicle and measurement frequency was set to 20 Hz. The measurement devices are displayed on the following Figure 27:

Figure 27 – Measurement devices (left: Leica CS20 controller, right: Leica GS18 smart antenna) (author)

Validation of the S.K.Neumanna crossing

The position of the camera is located at the roof of a high building which reduces the perspective distortion. There are also several places where drivers have to stop and give right of way which provides a good opportunity to measure a non-sudden braking. In the Figure 28, four calibration points are displayed. These points were measured with the GNSS station with a precision of 0.012 m. As the station provides data in a form of 3D deltas, the coordinates were converted to a local coordinate system. Z-coordinate is neglected as the road is assumed as a plane.

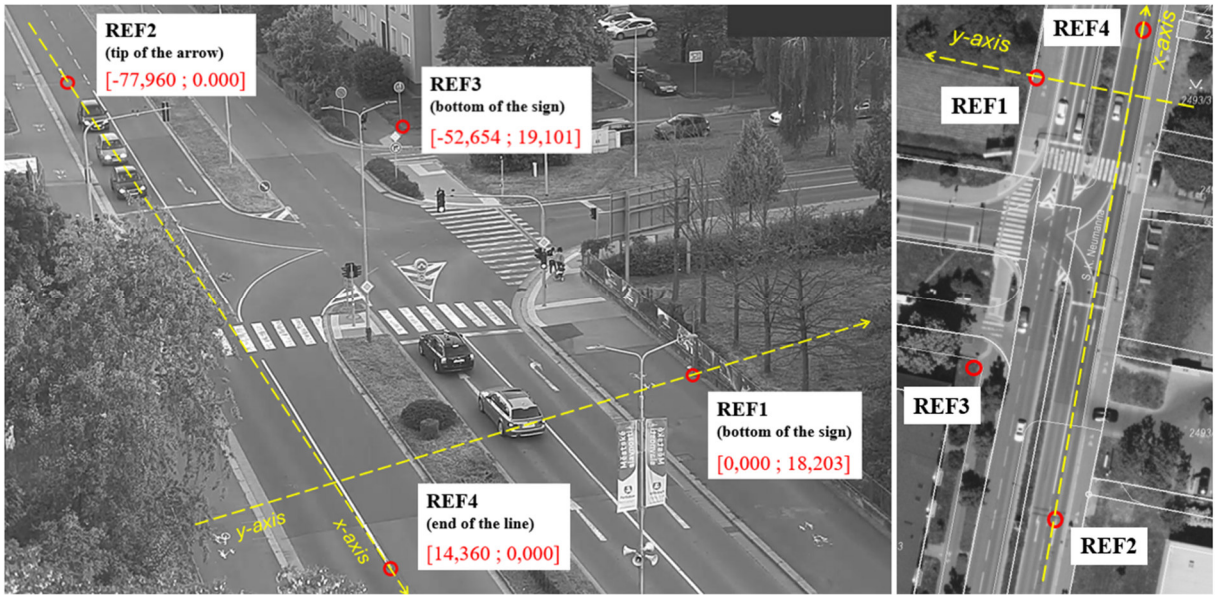


Figure 28 – Calibration configuration of the S.K.Neumann crossing (author)

Then, after obtaining the image-plane coordinates of the ground truth points, the homography matrix was generated. For the calibration, two methods of obtaining the homography matrix are used. First one estimates the homography matrix following the formulas stated in chapter 2.3.1 and then uses the non-linear optimization. The second one uses the built-in function cv.findHomography. Then the projection of image-plane points to the real-world coordinated is performed and difference against the ground truth points is calculated. Finally, the homography matrix with the lower error is used for the calibration.

The most precise version was the outcome from a python OpenCV function cv.findHomography. The ground truth points and their projection using the found homography matrix together with difference error is provided below.

$$ground\ truth = \begin{bmatrix} 0.000 & 18.203 \\ -77.960 & 0.000 \\ -52.654 & 19.101 \\ 14.360 & 0.000 \end{bmatrix} \tag{30}$$

$$projected\ points = \begin{bmatrix} -0.00000042 & 18.20299999 \\ -77.95999916 & -0.00000092 \\ -52.65399992 & 19.10099863 \\ 14.36000137 & -0.00000015 \end{bmatrix} \tag{31}$$

Hence, the differences between ground truth and projected image-plane points are:

$$differences = \begin{bmatrix} -0.00000042 & -0.00000001 \\ 0.00000084 & -0.00000092 \\ 0.00000008 & -0.00000137 \\ 0.00000137 & -0.00000015 \end{bmatrix} \quad (32)$$

According to the little, negligible, differences, the calibration process of the scene was successful.

4.2.4.2. Validation

The main purpose of the physical validation on the particular crossing was to be sure that the evaluated position of the vehicle is correct which was validated by comparing it with the data from GNSS. The GNSS data were transformed from global GPS coordinates to local, real-world coordinate system, presented in the previous calibration chapter.

Using the tracking SW, the test vehicle coordinates were obtained, extracted (as in each scene were usually 10-15 objects at the same time) and calibrated to the same local coordinate system. Measurements of all possible direction of movement of the vehicle were extracted and are displayed in the following overview (Figure 29).

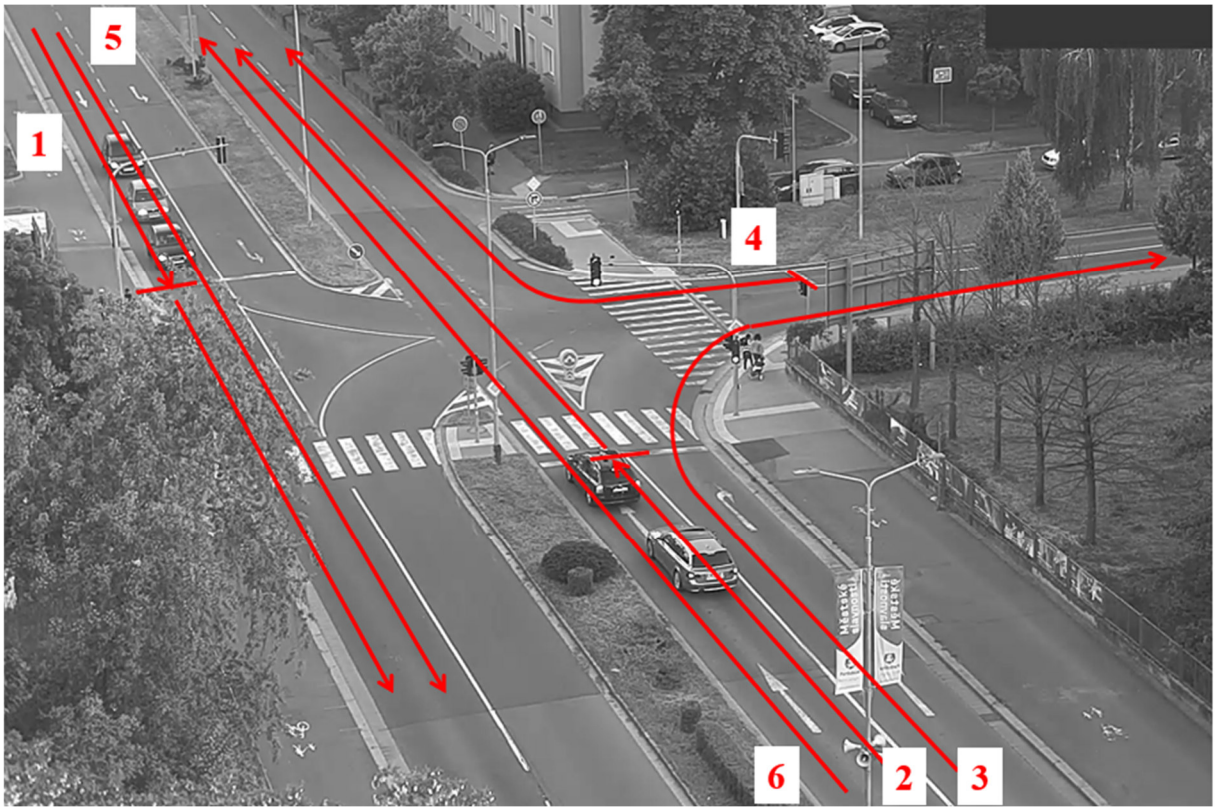


Figure 29 – Overview of validation measurements (author)

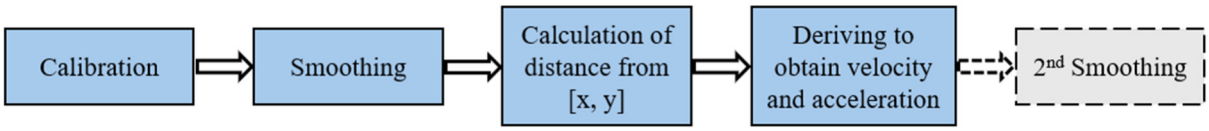
As can be seen in the Figure 29, six basic manoeuvres were selected for validation of the correct function of the tracker SW and later for data evaluation. In the Table 5 there is a description of the manoeuvres.

Table 5 – Description of manoeuvres evaluated for the SW validation.

Manoeuvre No.	Description
1	Direct drive. Deceleration, stop and acceleration on the traffic light.
2	Direct drive. Deceleration, stop and acceleration on the traffic light in the opposite way.
3	Direct drive and turning right.
4	Turning right and acceleration.
5	Direct constant drive.
6	Direct constant drive in the opposite way.

As a first result from the tracking SW, a time, x-axis and y-axis coordinates of the left-bottom, left-top, right-bottom and right-top points of bounding boxes were obtained. At first, a centre of each bounding box was calculated. The coordinates were in an image-plane coordinate system at this point warped by perspective distortion including considerable noise.

The following steps were performed to have a complete and processed set of data ready to be compared to the GNSS data. The order of the operations was important in order to avoid increasing any error from the raw data and was chosen as follows:



The calibration was performed as first because the smoothing shifts the points and calibration may scale a possible error of the correct but shifted points (homography matrix used for the calibration is non-linear). Moreover, the smoothing in the image plane can distort trajectories under perspective. Deriving must be done after the smoothing because each little noise at the distance level would be multiplied at the velocity and acceleration level. However, still a second round of smoothing is mostly needed to remove residual waves from the velocity and acceleration functions.

After the calibration step, points in the real-world space were obtained in the form of x- and y-axis coordinates in the local coordinate system. This section documents the trajectory smoothing strategies considered and implemented to refine the positional data, including:

1. Savitzky-Golay filtering,
2. Smoothing splines,
3. Zero-phase Butterworth low-pass filtering

The Savitzky–Golay (S-G) filter applies a least-squares polynomial fit of degree p within a moving window of width W (odd), estimating the central point. The smoothed value at sample i is:

$$\hat{y}_i = \sum_{j=-(W-1)/2}^{(W-1)/2} c_j y_{i+j} \tag{33}$$

where the convolution coefficients c_j depend on W and p and are derived analytically. This method preserves low-order moments (e.g., mean, slope) and is computationally efficient (Savitzky & Golay, 1964).

However, when applied to the particular data from the tracking SW:

- sampling frequencies varied and included missing time intervals, which meant that a fixed-width window could include data from large gaps,
- initial attempts with S-G filtering produced “bridges” across these gaps – unintended smoothing that litters the transitions and distorts the trajectory,
- additionally, at segment endpoints, the filter can introduce boundary artifacts unless special edge-handling strategies are used.

Due to these limitations, the S-G approach appeared to be unsuitable as a standalone method for smoothening the data from the tracking SW.

Smoothing splines estimate a function f(t) to minimize the penalized residual sum of squares:

$$min_f \left\{ \sum_{i=1}^{n} [y_i - f(t_i)]^2 + \lambda \int [f''(t)]^2 dt \right\} \quad (34)$$

where y_i are observed values (in this particular case the projected x-, y-coordinates), t_i a corresponding time point and $\lambda \geq 0$ is the smoothing parameter balancing fidelity (left term) and curvature penalty (right term).

As $\lambda \to 0$, f interpolates the data exactly; for large λ, f approaches a straight line. The solution yields a cubic spline with continuous second derivative (Reinsch, 1967; de Boor, 1978). This approach handles non-uniform time intervals naturally and avoids arbitrary window definitions, yielding smooth, differentiable trajectories ideal for motion analysis. (Craven & Wahba, 1979)

The zero-phase Butterworth low-pass filtering is defined by its transfer function, exhibiting maximally flat response in the passband. For an *n*-th order low-pass filter with cutoff frequency f_c, the continuous-time transfer function is:

$$H(s) = \frac{1}{\sqrt{1 + (s/\omega_c)^{2n}}} \quad (35)$$

where $\omega = 2\pi f_c$. In discrete form, at sampling frequency f_s, it is transformed via bilinear transform or digital filter design methods. Applying the filter with (forward and reverse) eliminates phase distortion. This approach preserves the signal alignment while attenuating high-frequency noise components. It also yields smooth output without introducing phase shift, which is critical for temporal trajectory integrity. To avoid smoothing across discontinuities (e.g., tracking gaps or missing intervals), the time series is partitioned at points where:

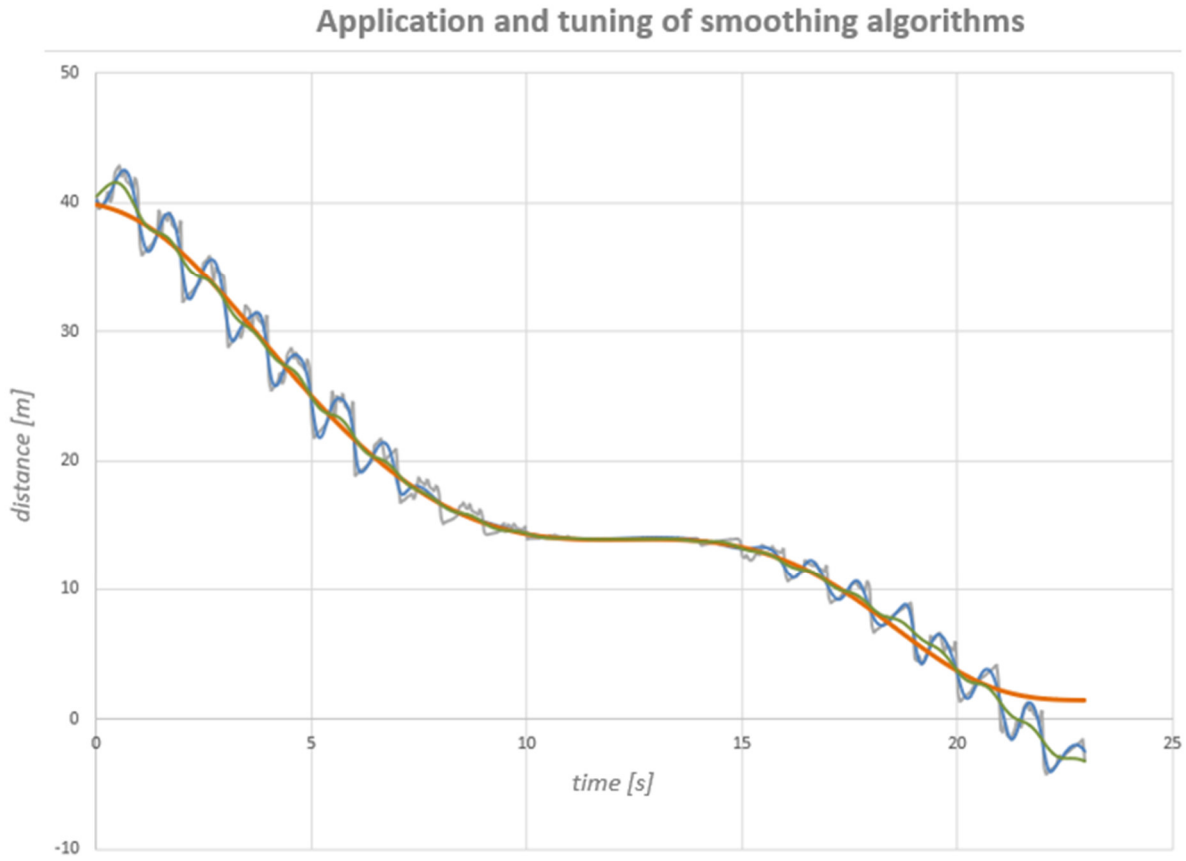
$$\Delta t_i = t_{i+1} - t_i > \alpha \cdot median(\Delta t) \quad (36)$$

with α as a gap factor (e.g. 5). Each contiguous segment is processed separately, and then the smoothed segments are recombined. This segmentation preserves data integrity and prevents artificial interpolation across gaps. (Gustaffson, F., 1996; Shumway, R.H. and Stoffer, D.S., 2017)

Initially, the Savitzky-Golay filter was preferred due to its simplicity and local fitting nature. However, it proved weaknesses in:

- Irregular sampling – requiring resampling to uniform grids or careful window adaptation.
- Gap interference – smoothing across large temporal gaps, falsely connecting disparate trajectory points.
- End effects – producing distortions at the edges without manual boundary handling.

In contrast, smoothing splines adapt naturally to uneven timestamps, provide global smoothness control via the smoothing parameter, and maintain continuity even in second derivatives, which is essential for trajectory stability. Similarly, the zero-phase Butterworth filter, while still based on windowing in frequency domain, offers strong noise rejection without temporal lag and can be applied segment-wise to respect data discontinuities. Therefore, the spline or Butterworth (zero-phase) approaches, combined with gap-aware segmentation, offer greater robustness and conceptual clarity for trajectory smoothing in the particular data context in this thesis. Also, a phase of tuning the smoothing algorithm parameters was very important, see an example in the Graph 5:



Graph 5 – Tuning the Butterworth smoothing algorithm (author)

In the Graph 5, various cut-off frequencies of Butterworth algorithm were used. Purple signal represents raw data, blue with application of 0.8 Hz, the optimum (orange) has a frequency of 0.15 Hz.

The next step was to obtain a distance travelled per each time step. Firstly, a distance increment per each time step was calculated using the following formula (basically an Euclidean distance):

$$dist_{increment} = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \tag{37}$$

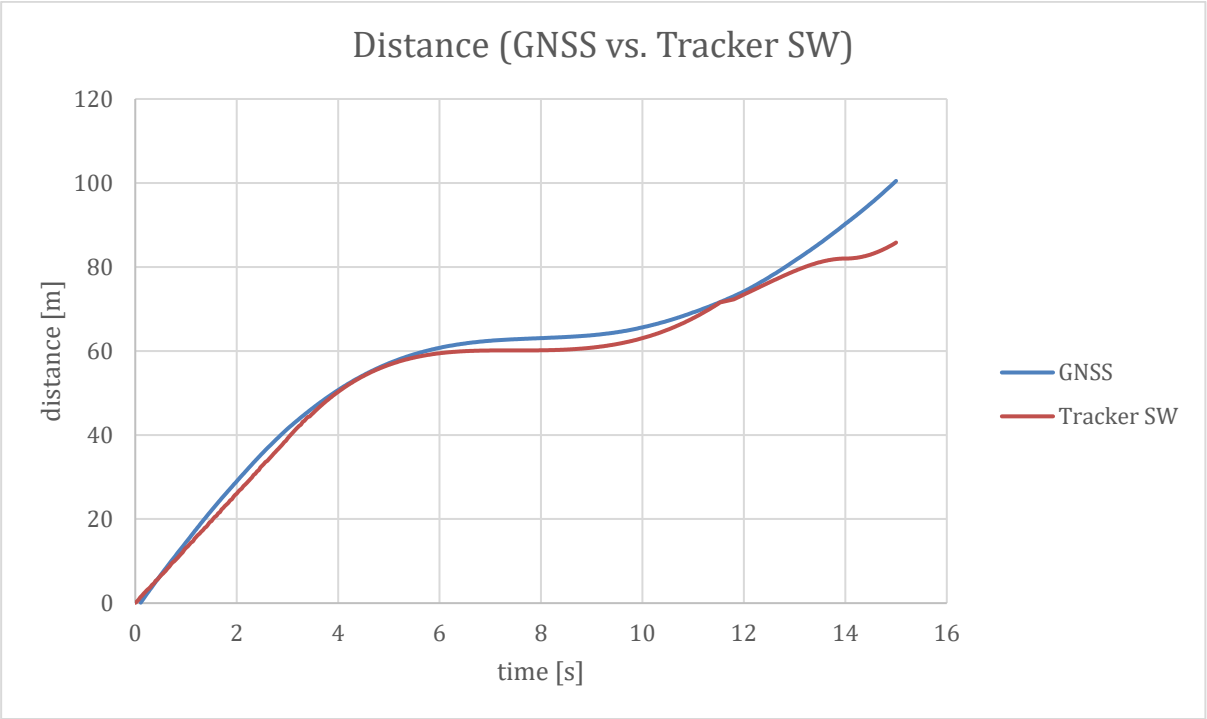
A total distance travelled per each time moment is a cumulative function of the $dist_{increment}$.

Velocity and acceleration were derived as the first and second derivatives of the distance with respect to time.

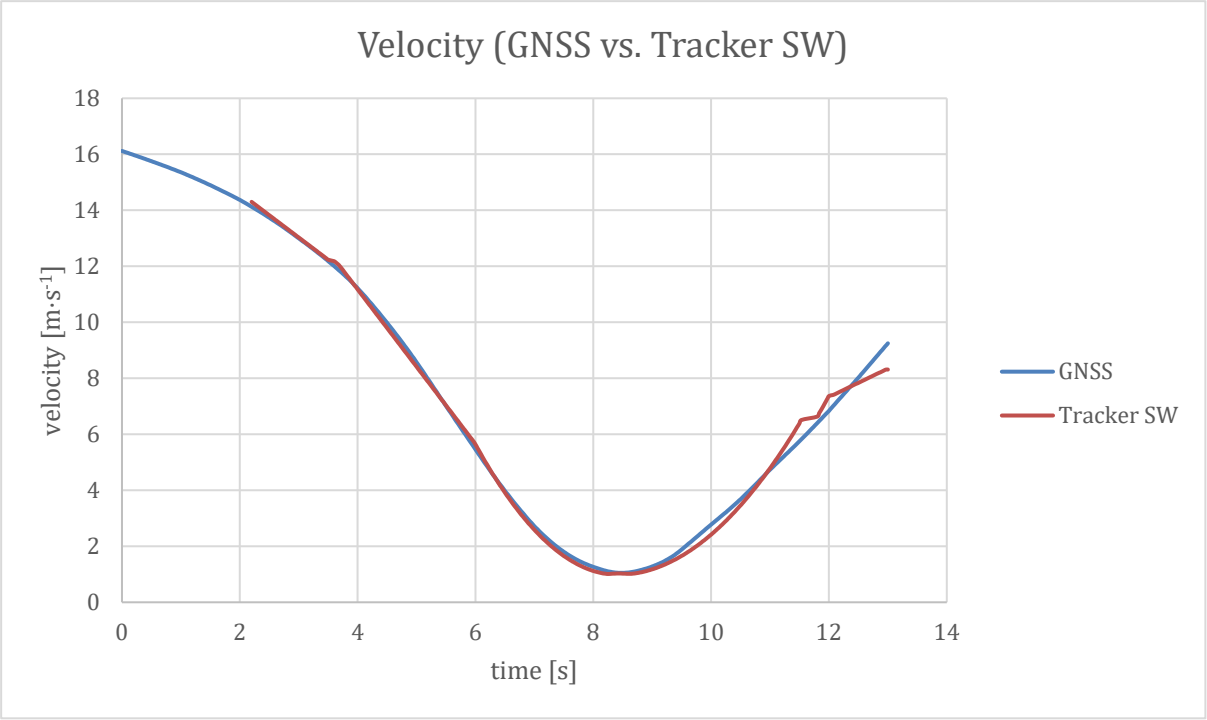
$$v(t) = \frac{ds(t)}{dt} \tag{38}$$

$$a(t) = \frac{dv(t)}{dt} \tag{39}$$

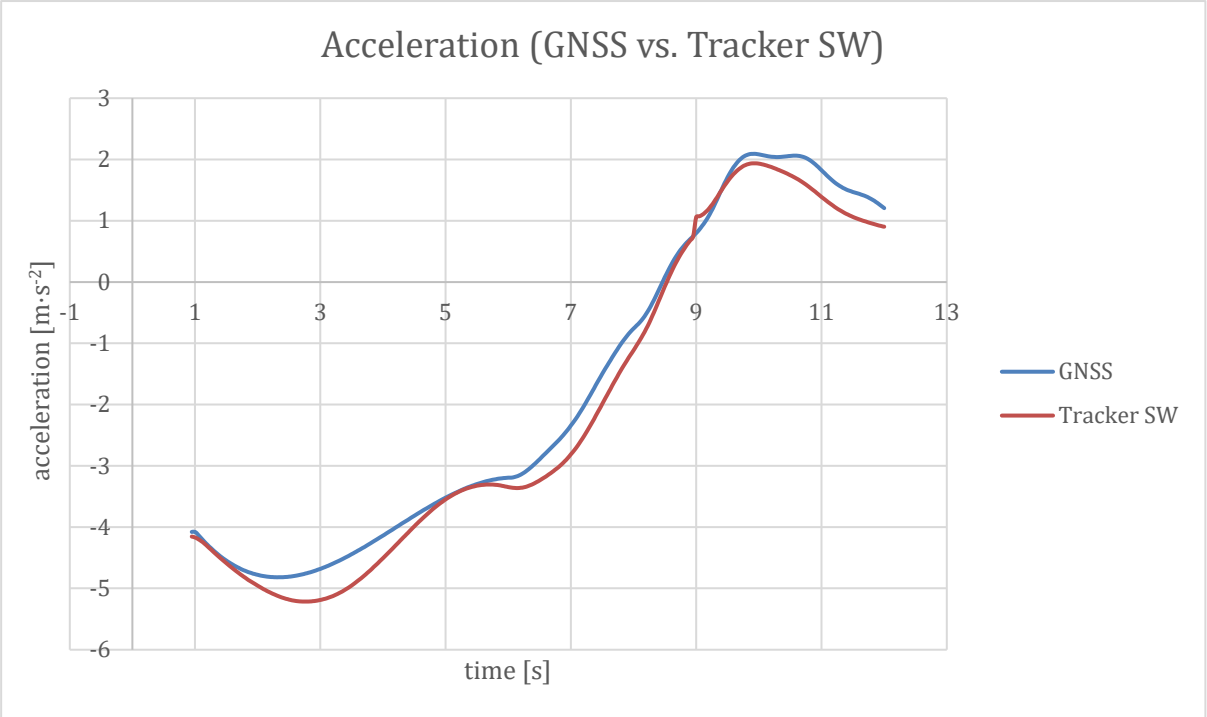
After obtaining the velocity and acceleration with respect to time, another round of smoothing had to be performed as some residual mistakes in the distance function were amplified. Below graphs 6-8 of the distance, velocity and acceleration can be seen, obtained from the GNSS and tracker SW and processed. This is a representation of the drive number 2. The complete set of the graphs for all the six drives is attached to the thesis as an Appendix 1.



Graph 6 – Distance evaluation from GNSS and tracker SW.



Graph 7 – Velocity evaluation from the GNSS and tracker SW



Graph 8 – Acceleration evaluation from the GNSS and tracker SW

In the distance graph, the curve from tracker SW precisely follows the GNSS curve. Only at the end the tracker SW curve changes direction which is caused by disappearing of the vehicle in the perspective. The same case is the velocity graph. For the acceleration, there is a little irregular offset visible around ca. 0.5 $m \cdot s^{-2}$. It can be caused by the fact that the tracker SW needs more time to stabilize the bounding box after detecting the vehicle for the first time. In this scene, the vehicle braked intensively only several seconds after appearing in the scene which resulted in higher error from the beginning. Considering the GNSS and tracker SW curves from all the validation drives, the SW is precise enough after stabilization of the tracking algorithm to obtain reliable results.

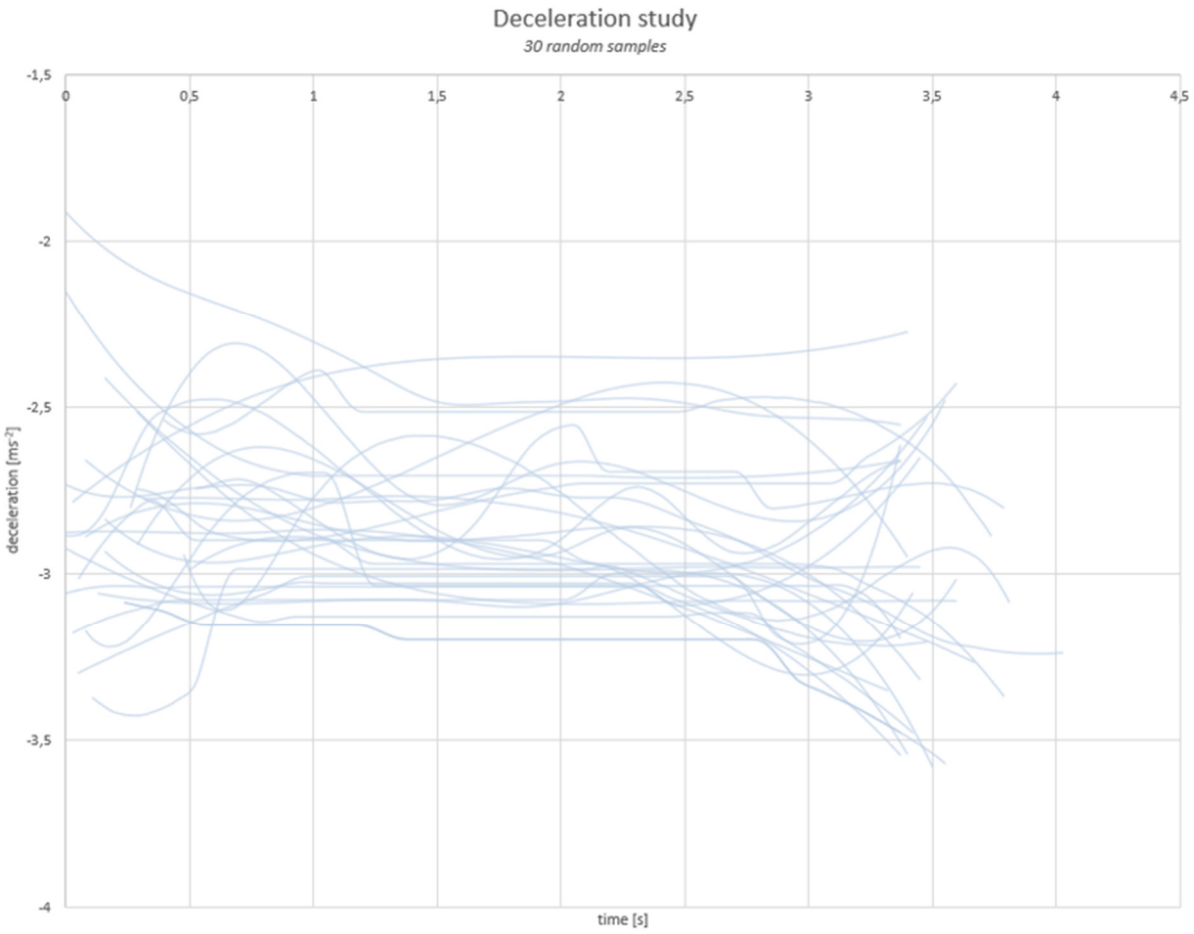
4.2.5. Data evaluation

In the road accident expert's praxis, there are quantities, which values are still only empirically estimated and aren't based on deep research. Among such quantities belongs for example a non-sudden braking (expressed in deceleration). There are cases, when the non-sudden deceleration is necessary to be used, mostly when the experts must determine, whether the driver would have stopped in front of some obstacle by braking normally within his comfort zone. The non-sudden braking can be extracted for example from such situations, when drivers arrive to red traffic lights, when they stop in front of a pedestrian crossing to let pedestrians go across the road or when they must give a right of way.

As it was already mentioned in the beginning of the thesis, a method of independent observation and then statistical evaluation of the results was chosen. The advantage of this approach is that the subjects of measurements (drivers) were not aware that they are being measured and therefore it is ensured that their behaviour was natural. The measurement was conducted on the S.K.Neumanna x Pichlova crossing in Pardubice as the scene has already been validated by comparison with the GNSS record. Moreover, only data from the straight part of the road were used for evaluation as the turn at the right is partially covered in the scene.

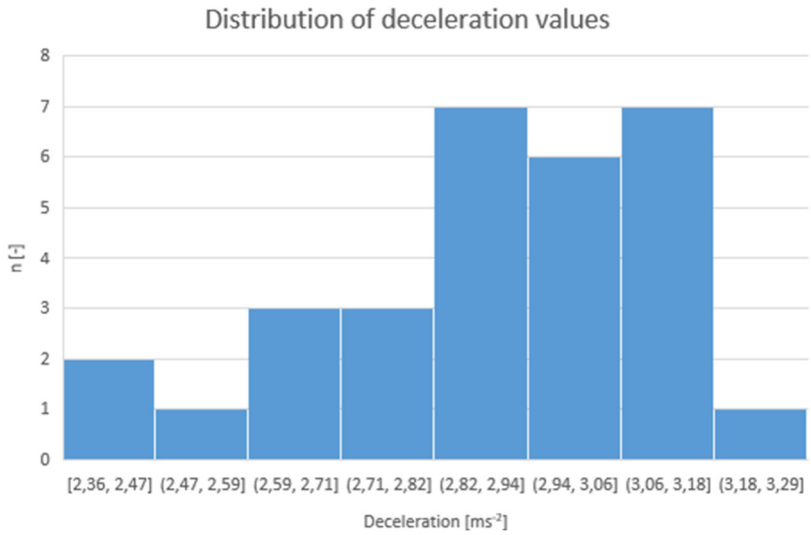
For the purpose of non-sudden braking measurement an already calibrated and validated scene was used. The video records from the crossing were cut to multiple representative samples where there is a good visibility of the measured vehicle, no occlusion and the behaviour of the vehicle is not unusual. Then, the video records were automatically processed by the tracking SW, the raw data were automatically smoothened the same way as it was performed in the validation phase and then the mean deceleration of the braking part was obtained.

As it was already mentioned in the chapter 4.2.1 Input data, 30 independent and representative deceleration values were obtained and statistically evaluated. In order to exclude transitional events from the evaluation and let the tracker stabilize after a detection, data were evaluated after 1 s after the detection. In the graph 9 below, deceleration curves are provided:



Graph 9 – Output deceleration curves from 30 measurements (author)

A distribution of means of individual decelerations is displayed in the graph 10:



Graph 10 – Deceleration histogram

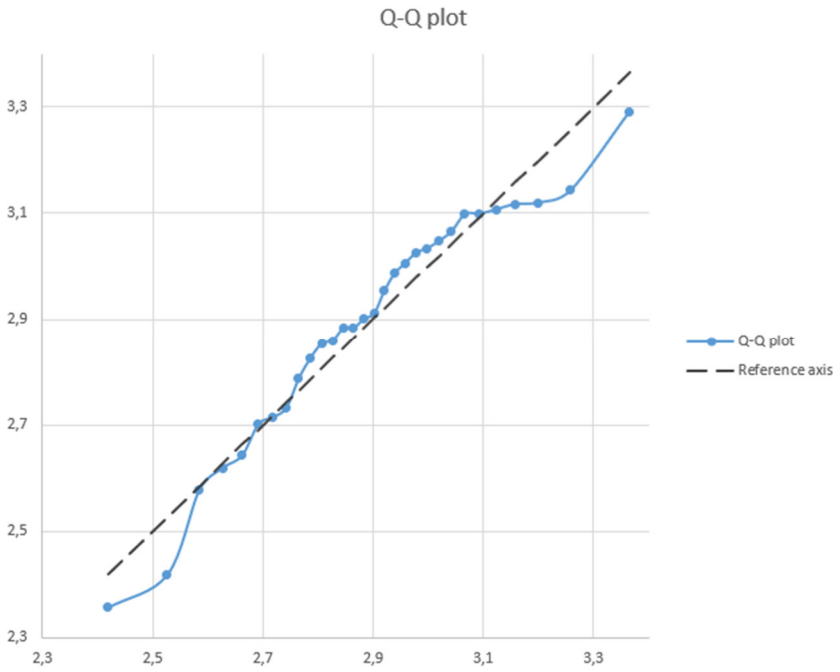
As the character of obtained data is supposed to be normally distributed, a normality test was performed. A combination of visual assessment Q-Q plot and Shapiro-Wilk test was used for the sample size of 30 measurements.

Q-Q plot

A Quantile–Quantile (Q–Q) plot is a graphical method used to assess whether a dataset follows a particular theoretical distribution. The procedure involves first sorting the observed data in ascending order and then assigning each value a corresponding theoretical quantile, typically calculated as $(i - 0.5)/n$, where i is the rank of the observation and n is the total sample size. These probabilities are then transformed into expected quantile values of the target distribution using the inverse cumulative distribution function (in this particular case, NORM.S.INV or NORM.INV for the normal distribution). When the data are approximately normally distributed, the plotted points will align closely with a straight line.

The strength of the Q–Q plot lies in its simplicity and its ability to highlight deviations from normality, particularly in the distribution tails where formal statistical tests may have limited sensitivity. For this reason, Q–Q plots are often used in combination with formal normality tests, such as the Shapiro–Wilk test, to provide both a visual and a statistical assessment of distributional assumptions (Field, 2013; Wilk and Gnanadesikan, 1968; Thode, 2002).

In the graph 11 below, a Q-Q plot for the obtained data is provided:



Graph 11 – Q-Q plot of the obtained data (author)

It is visible in the Q-Q plot that the data are mostly aligned with the diagonal axis of the graph which indicates normal distribution. Only in the beginning and the end there are outlying values that are slightly differentiating from the axis.

Shapiro-Wilk test

The Shapiro-Wilk test is one of the most widely used formal statistical tests for assessing whether a sample comes from a normally distributed population. It was first introduced by Shapiro and Wilk (1965) and has since become a standard method, particularly for small to medium-sized datasets (typically n < 50), though it can be applied to larger samples as well. The test statistic, denoted as *W*, is based on a ratio that compares the squared correlation between the ordered sample values and the corresponding expected values from a normal distribution. Values of *W* close to 1 indicate that the data are consistent with normality, whereas smaller values suggest deviations from normality.

One of the main advantages of the Shapiro-Wilk test is its relatively high statistical power compared to alternative methods such as the Kolmogorov-Smirnov test, especially when detecting departures from normality in the tails of the distribution (Razali and Wah, 2011; Thode, 2002). In practice, the test provides both the statistic *W* and an associated *p*-value. If the *p*-value is greater than a chosen significance level (e.g. α = 0.05), the null hypothesis of normality is not rejected, implying that the data do not significantly deviate from a normal distribution. Due to its reliability and sensitivity, the Shapiro-Wilk test is recommended as the primary test of normality in many statistical applications.

The Shapiro–Wilk statistic is calculated as follows:

$$W = \frac{\left(\sum_{i=1}^{n} a_i x_{(i)}\right)^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{40}$$

where $x_{(i)}$ are the ordered sample values, $\bar{x}$ is the sample mean, and the coefficients a_i are constants derived from the expected values of order statistics of a normally distributed sample and their covariance matrix (Shapiro and Wilk, 1965). The numerator represents the squared weighted sum of the ordered values, while the denominator is the total sample variance. Decision criteria are based on the I-value associated with *W*: if $p \leq \alpha$, the null hypothesis of normality is rejected; if $p > \alpha$, the assumption of normality is considered plausible. (Montgomery & Runger, 2014)

After performing the test on the particular data, The Shapiro-Wilk statistic $W = 0.9561$ and p-value = 0.2451. Normality was assessed using the Shapiro–Wilk test at a significance level

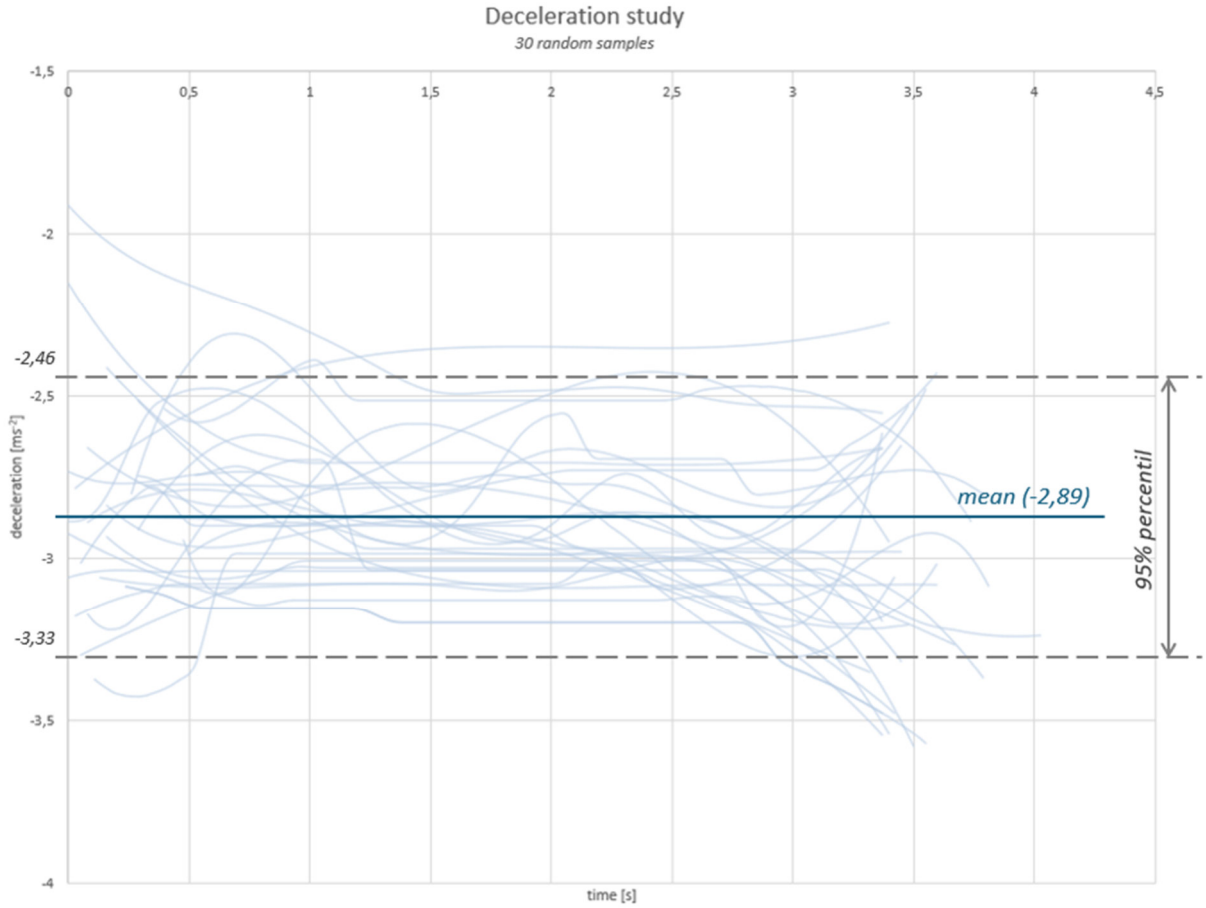
of $\alpha = 0.05$. The test returned a p-value of 0.25, which is greater than 0.05; therefore, the null hypothesis of normality could not be rejected, and the data were considered normally distributed.

As a conclusion, the deceleration measurements can be summarized as follows in the Table 6:

Table 6 – Summary

Quantity	Value
Mean	-2.89
Standard deviation	0.22
Central 95% interval	(-3.33; -2.46)

In the following Graph 12, the evaluated data are highlighted in the deceleration graph.



Graph 12 – Deceleration graph with highlighted statistical values (author)

5. CONCLUSION

In the presented thesis a complex procedure was developed, building a backbone of automatic bulk data analysis from traffic surveillance cameras. The pipeline consists of obtaining input data from traffic surveillance cameras and preparation of scenes as an input for the further steps. The core of the automatic data acquisition is a synthesis of a detection and tracking SW based on a pre-trained CNNs (YOLOv7 + DeepSORT) and their implementation using a suitable dataset for learning (MS COCO). A crucial part in terms of precision is a correct calibration of the scene, using a PnP principle which proved to be versatile and precise.

In order to evaluate the precision and application limits of the SW and calibration method, a validation through simulation and physical measurement was performed. As a first step, drives of a vehicle in a rendered video from simulation under various controlled conditions were processed and evaluated. In the second step, kinematic and dynamic quantities were evaluated from a real measurement on a selected crossing from both the GNSS station and tracker SW and compared together. After a successful validation, a non-sudden braking representing a quantity frequently used in a traffic accidents analysis praxis was evaluated from a random sample from a validated traffic surveillance camera.

The main objectives of the thesis were fulfilled:

- A complex and universal method for not only traffic analysis was developed.
- Its correct application and precision were proven on particular samples of video records from a real crossing.
- The correct functionality and boundaries of the method were determined by two-step validation through a simulation and physical measurement.
- On a validated scene, a non-sudden braking quantity was obtained and evaluated.

Following the critical path for obtaining the non-sudden braking deceleration, the scene has been calibrated creating a homography matrix perfectly fitting the real road plane. As the next step, 30 samples of video shots of random vehicles braking to stop were processed through the tracking SW. The data were smoothened and statistically evaluated, showing that the **non-sudden braking deceleration lies between -3.33 and -2.46 m·s^{-2} (95 % central interval) with a mean of -2.89 m·s^{-2}.**

Relating the evaluated results to a recent forensic praxis of traffic accidents analysis, the result is aligned with the interpretation that the non-sudden braking lies around the value of a half of the minimum required mean fully developed deceleration (MFDD) for passenger cars

5.8 $m \cdot s^{-2}$, hence 2.9 $m \cdot s^{-2}$ according to the UN/ECE Regulation No. 13-H, and Czech technical regulations (Vyhláška č. 341/2014 Sb.).

The results of the thesis can be used directly in the traffic accidents analysis praxis as a reference value when evaluating the suddenness of an event, which leads to a more precise understanding of the accident configuration and driver's behaviour before it. The method itself is versatile and with minor adjustments can be applied on a variety of cases when an automatic object detection and tracking together with a robust calibration method is needed. Moreover, various proposals on further research are introduced. The practical use of the outcome of the thesis as well of the method itself together with many directions of possible further development highlight the perspectivity of the research.

The initial hypothesis regarding the capability of the developed tracking system to provide sufficiently precise kinematic and dynamic parameters of traffic users has been confirmed under specific circumstances. To achieve a reliable precision, the position of the camera must fulfil rules defined in the thesis. At the same time, the road must be flat, there should be maximally a little occlusion of objects and the movements of objects that are analysed has to last at least 1 second as the SW needs to stabilize after detection. The hypothesis would be confirmed fully (no constraints of the applicability of the proposed system) if wider research of the detection and tracking SW is performed resulting in more precise and robust detection and tracking with lower computational power demand at the same time.

6. FURTHER RESEARCH PROPOSAL

The thesis introduces a complex methodology of obtaining a specific kind of data. Following the pipeline in the Figure 15 of the procedure, there were considered several assumptions which enabled to obtain a robust method and desired results in sufficient precision. Below in this chapter, a list of further research proposals is provided. Conducting and developing of the following research can significantly contribute to the field of automated traffic analysis as well as generally automatic object detection and tracking.

- Evaluating also other kinematic and dynamic quantities (e.g. safe pass time, natural acceleration of drivers, time threshold of braking/accelerating when an orange signal appears on the traffic light etc.).
- Difference between multiple types of cars, year seasons, illumination conditions, time, cities, gender.
- Correction of the offset in the graphs.
- Calibration using aerial images and its precision.
- Error distribution across the whole procedure.
- Correction of the mismatch of positions of the centre of bounding boxes (optical centre of the object in the video point of view and a real mass centre of the object).

Furthermore, it could contribute to the procedure reliability and robustness, if the FMEA or similar analysis would be performed and recommended actions would be implemented.

7. REFERENCES

Addala, S., 2020. *Research paper on vehicle detection and recognition* [online]. Available at: https://www.researchgate.net/publication/344668186_Research_paper_on_vehicle_detection_and_recognition [Accessed 20 August 2025].

Adžemović, M., 2025. *Deep Learning-Based Multi-Object Tracking: A Comprehensive Survey from Foundations to State-of-the-Art* [online]. Preprint (arXiv:2506.13457). Available at: https://arxiv.org/abs/2506.13457 [Accessed 20 August 2025].

Ali, M.H., Abu Talib, M., Abualigah, L., Rashid, M., and Elaziz, M.A., (2024). *A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS. Journal of Imaging*, 10(2), p.27. https://doi.org/10.3390/jimaging10020027

Alom, M.Z. and Taha, T.M., 2017. *Robust multi-view pedestrian tracking using neural networks* [online]. arXiv preprint arXiv:1704.06370. Available at: https://arxiv.org/abs/1704.06370 [Accessed 20 August 2025].

Arafat, M.Y., Khairuddin, A.S.M., Khairuddin, U. and Paramesran, R., 2019. *Systematic review on vehicular licence plate recognition framework in intelligent transport systems. IET Intelligent Transport Systems*, 13(5), pp.745-755. https://doi.org/10.1049/iet-its.2018.5341

Aris, N.A.M., Jamaian, S.S., and others, 2025. *High-accuracy vehicle detection in different traffic densities using improved Gaussian mixture model with cuckoo search optimization*. *International Journal of Advanced Computer Science and Applications*, 16(1), pp.1039–1048. Available at: https://doi.org/10.14569/IJACSA.2025.01601100

Arya, M.C. and Rawat, A., 2020. *A review on YOLO (You Look Only One) – an algorithm for real time object detection*. *Journal of Engineering Sciences,* 11, pp.554–557.

Barla, N., 2022. *The complete guide to object tracking [+V7 tutorial]* [online]. V7, 21 October 2022. Available at: https://www.v7labs.com/blog/object-tracking-guide [Accessed 5 December 2022].

Barnich, O. and Van Droogenbroeck, M., 2009. *ViBe: A universal background subtraction algorithm for video sequences*. *IEEE Transactions on Image Processing* 20(6), pp.1709–1724. https://doi.org/10.1109/TIP.2010.2101613

Bartl, V., Špaňhel, J., Dobeš, P., Juránek, R. and Herout, A., 2021. *Automatic camera calibration by landmarks on rigid objects*. *Machine Vision and Applications*, 32(1), p.2. https://doi.org/10.1007/s00138-020-01125-x

Behrendt, K., 2019. *Boxy vehicle detection in large images*. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*. IEEE, pp. [online]. Available at: https://openaccess.thecvf.com/content_ICCVW_2019/html/CVRSUAD/Behrendt_Boxy_Vehicle_Detection_in_Large_Images_ICCVW_2019_paper.html [Accessed 20 August 2025].

Bernardin, K., Elbs, A. and Stiefelhagen, R., 2006. *Multiple object tracking performance metrics and evaluation in a smart room environment*. In: *Proceedings of the Sixth IEEE International Workshop on Visual Surveillance (in conjunction with ECCV)*, Vol. 90. Citeseer, pp. [online]. Available at: https://www.semanticscholar.org/paper/Multiple-Object-Tracking-Performance-Metrics-and-in-Bernardin-Elbs/176d4c1bd97f011567d55e5fa9f875c671dedc60 [Accessed 20 August 2025].

Berg, J., Jilek, P., Pokorný, J. and Krmela, J., 2022. *Metody předzpracování obrazu pro automatickou detekci účastníků silničního provozu. Perner's Contacts*, 17(2). doi:10.46585/pc.2022.2.2389. ISSN 1801-674X.

Bewley, A., Ge, Z., Ott, L., Ramos, F. and Upcroft, B., 2016. *Simple online and realtime tracking*. In: *2016 IEEE International Conference on Image Processing (ICIP)*, pp.3464–3468. IEEE. https://doi.org/10.1109/ICIP.2016.7533003

Bhardwaj, R., Tummala, G.K., Ramalingam, G., Ramjee, R. and Sinha, P., 2018. *AutoCalib: automatic traffic camera calibration at scale*. *ACM Transactions on Sensor Networks (TOSN)*, 14(3-4), pp.1–27. https://doi.org/10.1145/3199667

Bochkovskiy, A., Wang, C.-Y. and Liao, H.-Y.M., 2020. *YOLOv4: optimal speed and accuracy of object detection* [online]. arXiv preprint arXiv:2004.10934. Available at: https://arxiv.org/abs/2004.10934 [Accessed 20 August 2025].

de Boor, C., 1978. *A practical guide to splines*. Revised edition. New York: Springer.

Brouwers, G.M.Y.E., Zwemer, M.H., Wijnhoven, R.G.J. and de With, P.H.N., 2016. Automatic calibration of stationary surveillance cameras in the wild. In: *Computer Vision – ECCV 2016 Workshops*, Lecture Notes in Computer Science, vol. 9914, pp. 743–759. Springer, Cham. Available at: https://doi.org/10.1007/978-3-319-48881-3_52

Bradáč, A., 1997. *Soudní inženýrství*. Brno: Akademické nakladatelství CERM, s.r.o. ISBN 80-7204-057-X.

Bradáč, A., Semela, M., Kledus, R., Bilík, M., Křižák, M., Mikulec, R., Bucsuházy, K., Belák, M., Mičunek, T., Frydrýn, M., Nouzovský, L., Svatý, Z., Radová, Z., Fryšták, M., Vojtíšek, T., Pokorný, J. and **Berg, J.**, 2021. *Teorie a praxe analýzy silničních nehod*. Brno: VUTIUM. ISBN 978-80-214-6209-0. doi:10.13164/usi.book.tpasn.

Cai, S. and Wu, Z., 2018. *Camera calibration with coplanar conics: a unified explanation and ambiguity analysis*. *IPSJ Transactions on Computer Vision and Applications*, 10, p.5. https://doi.org/10.1186/s41074-018-0050-y

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A. and Zagoruyko, S., 2020. *End-to-end object detection with transformers*. In: A. Vedaldi, H. Bischof, T. Brox and J-M. Frahm, eds. *Computer Vision – ECCV 2020*. Lecture Notes in Computer Science, vol.12346. Cham: Springer, pp.213–229. https://doi.org/10.1007/978-3-030-58452-8_13

Courthoud, M., 2022. *How to compare two or more distributions* [online]. *Towards Data Science*. Available at: https://towardsdatascience.com/how-to-compare-two-or-more-distributions-9b06ee4d30bf [Accessed 6 December 2022].

Craven, P. and Wahba, G., 1978. *Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross-validation*. *Numerische Mathematik*, 31(4), pp.377–403. https://doi.org/10.1007/BF01404567.

Česká republika. Ministerstvo dopravy, 2000. *Zákon č. 361/2000 Sb., o provozu na pozemních komunikacích*. Sbírka zákonů, částka 101/2000. Praha: Ministerstvo dopravy ČR.

Česká republika. Ministerstvo dopravy, 2014. *Vyhláška č. 341/2014 Sb., o schvalování technické způsobilosti a o technických podmínkách provozu vozidel na pozemních komunikacích.* Sbírka zákonů, částka 134/2014. Praha: Ministerstvo dopravy ČR.

ČÚZK, n.d. *Nahlížení do katastru nemovitostí* [online]. Český úřad zeměměřický a katastrální (ČÚZK). Available at: https://nahlizenidokn.cuzk.cz [Accessed 20 August 2025].

Dai, J., Li, Y., He, K. and Sun, J., 2016. *R-FCN: object detection via region-based fully convolutional networks*. In: *Advances in Neural Information Processing Systems (NeurIPS 2016)*, 29. Curran Associates, Inc. pp.379–387.

Do, V.H., Nghiem, L.H., Thi, N.P. and Ngoc, N.P., 2015. *A simple camera calibration method for vehicle velocity estimation*. In: *2015 12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*. IEEE, pp.1–5. doi: 10.1109/ECTICon.2015.7207027

Dong, H., Wen, M. and Yang, Z., 2019. *Vehicle speed estimation based on 3D ConvNets and non-local blocks*. *Future Internet*, 11(6), p.123. https://doi.org/10.3390/fi11060123

Dubská, M., Herout, A. and Sochor, J., 2014. *Automatic camera calibration for traffic understanding*. In: *Proceedings of the British Machine Vision Conference (BMVC 2014)*. BMVA Press, pp.1–12. doi: 10.5244/C.28.42.

Eddy, C., De Saxe, C. and Cebon, D., 2018. *Camera-based measurement of cyclist motion*. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 233(7), pp.1793–1805. https://doi.org/10.1177/0954407018777164

Felzenszwalb, P.F. and Huttenlocher, D.P., 2004. *Efficient graph-based image segmentation*. *International Journal of Computer Vision*, 59(2), pp.167–181. https://doi.org/10.1023/B:VISI.0000022288.19776.77

Feng, H., Shi, W., Chen, F., Byon, Y.J., Heng, W. and Pan, S., 2020. A calculation method for vehicle movement reconstruction from videos. *Journal of Advanced Transportation*, 2020, Article ID 8896826. Available at: https://doi.org/10.1155/2020/8896826

Fernández Llorca, D., Hernández Martínez, A. and García Daza, I., 2021. Vision-based vehicle speed estimation: a survey. *IET Intelligent Transport Systems*, 15(8), pp. 987–1005. Available at: https://doi.org/10.1049/itr2.12079

Field, A., 2013. *Discovering Statistics Using IBM SPSS Statistics.* 4th ed. London: Sage.

Filipiak, P., Golenko, B. and Dolega, C., 2016. NSGA-II based auto-calibration of automatic number plate recognition camera for vehicle speed measurement. In: Squillero, G. and Burelli, P., eds. *Applications of Evolutionary Computation. EvoApplications 2016*. Lecture Notes in Computer Science, vol. 9597, pp. 803–818. Cham: Springer. Available at: https://doi.org/10.1007/978-3-319-31204-0_51

Gaikwad, M., 2022. *Deep SORT: simple online and real-time tracking with deep associative metric*. *Medium*, 15 June. Available at: https://medium.com/@mohit_gaikwad/deep-sort-simple-online-and-real-time-tracking-with-deep-associative-metric-94138d528ff1 [Accessed 12 August 2025].

Galliot, 2022. *Camera Calibration Using Homography Estimation*. Available at: https://galliot.us/blog/camera-calibration-using-homography-estimation/. [Accessed: 21 August 2025].

Geiger, A., Lenz, P. and Urtasun, R., 2012. *Are we ready for autonomous driving? The KITTI vision benchmark suite*. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp.3354–3361. https://doi.org/10.1109/CVPR.2012.6248074

Giannakeris, P., Kaltsa, V., Avgerinakis, K., Briassouli, A., Vrochidis, S. and Kompatsiaris, I., 2018. Speed estimation and abnormality detection from surveillance cameras. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW 2018)*, pp. 93–99. IEEE. Available at: https://doi.org/10.1109/CVPRW.2018.00020

Girshick, R., 2015. *Fast R-CNN*. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV 2015)*. IEEE, pp.1440–1448. https://doi.org/10.1109/ICCV.2015.169

Girshick, R., Donahue, J., Darrell, T. and Malik, J., 2014. *Rich feature hierarchies for accurate object detection and semantic segmentation*. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2014)*. IEEE, pp.580–587. https://doi.org/10.1109/CVPR.2014.81

Guillou, E., Meneveaux, D., Maisel, E. and Bouatouch, K., 2000. *Using vanishing points for camera calibration and coarse 3D reconstruction from a single image*. *The Visual Computer*, 16(7), pp.396–410. https://doi.org/10.1007/PL00013394

Gunawan, A.A.S., Tanjung, D.A. and Gunawan, F.E., 2019. *Detection of vehicle position and speed using camera calibration and image projection methods*. *Procedia Computer Science*, 157, pp.255–265. Available at: https://www.sciencedirect.co$^{m/s}$cience/article/pii/S187705091931083X [Accessed 20 August 2025].

Gustafsson, F., 1996. *Determining the initial states in forward-backward filtering*. *IEEE Transactions on Signal Processing*, 44(4), pp.988–992. https://doi.org/10.1109/78.492552

Hata, K. and Savarese, S., n.d. *CS231A Course Notes 1: Camera Models*. Stanford University [online]. Available at: https://web.stanford.edu/class/cs231a/course_notes/01-camera-models.pdf [Accessed 5 December 2022].

He, K., Gkioxari, G., Dollár, P. and Girshick, R., 2017. *Mask R-CNN*. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV 2017)*. IEEE, pp.2961–2969. https://doi.org/10.1109/ICCV.2017.322

Held, D., Thrun, S. and Savarese, S., 2016. *Learning to track at 100 fps with deep regression networks*. In: *European Conference on Computer Vision (ECCV 2016)*. Springer, Cham, pp.749–765. https://doi.org/10.1007/978-3-319-46448-0_45

Huang, L., Wu, J., Zhang, R., Zhao, D. and Wang, Y., 2020. *Comparative analysis & modelling for riders' conflict avoidance behavior of E-bikes and bicycles at un-signalized intersections*. *IEEE Intelligent Transportation Systems Magazine*, 13(4), pp.131–145. https://doi.org/10.1109/MITS.2019.2926272

Huang, T., Ogasawara, G. and Russell, S., 1993. *Symbolic traffic scene analysis using dynamic belief networks*. California PATH Research Report. Institute of Transportation Studies, University of California, Berkeley.

Chaabane, M., Zhang, P., Beveridge, J.R. and O'Hara, S., 2021. *Deft: Detection embeddings for tracking*. arXiv preprint arXiv:2102.02267. Available at: https://arxiv.org/abs/2102.02267 [Accessed 20 August 2025].

Chai, J., Zeng, H., Li, A. and Ngai, E.W., 2021. *Deep learning in computer vision: A critical review of emerging techniques and application scenarios*. *Machine Learning with Applications*, 6, p.100134. https://doi.org/10.1016/j.mlwa.2021.100134

Chatterjee, C.C., 2019. Basics of the classic CNN. *Towards Data Science* [online], 31 July 2019. Available at: https://medium.com/data-science/basics-of-the-classic-cnn-a3dce1225add [Accessed 22 August 2025].

Chen, S., Sun, P., Song, Y. and Luo, P. 2022. *DiffusionDet: Diffusion model for object detection*. ArXiv preprint arXiv: 2211.09788. https://doi.org/10.48550/arXiv.2211.09788.

Itu, R. and Danescu, R.G., 2020. *A self-calibrating probabilistic framework for 3D environment perception using monocular vision*. Sensors, 20(5), p.1280. https://doi.org/10.3390/s20051280.

Jiang, P., Ergu, D., Liu, F., Cai, Y. and Ma, B., 2022. *A review of YOLO algorithm developments*. Procedia Computer Science, 199, pp.1066–1073. https://doi.org/10.1016/j.procs.2022.01.135.

Johansson, M., Laureshyn, A. and Nilsson, M., 2020. *Video analysis of pedestrian movement (VAPM) under different lighting conditions—Method exploration*. Energies, 13(16), p.4141. https://doi.org/10.3390/en13164141.

Jung, I., Son, J., Baek, M. and Han, B., 2018. Real-time MDNet. *In: Proceedings of the European Conference on Computer Vision (ECCV 2018)*. Cham: Springer, pp.83–98. Available at: http://openaccess.thecvf.com/content_ECCV_2018/html/Ilchae_Jung_Real-Time_MDNet_ECCV_2018_paper.html

Juránek, R., Špaňhel, J., Sochor, J., Herout, A. and Novák, J., 2019. *Visual analysis of vehicle trajectories for determining cross-sectional load density*. *Transactions on Transport Sciences*, 10(1), pp.50–57. https://doi.org/10.5507/tots.2019.002

Karmann, K.-P. and von Brandt, A., 1990. *Moving object recognition using an adaptive background memory.* In *Time-Varying Image Processing and Moving Object Recognition*, Vol. 2. Amsterdam: Elsevier, pp. 297–307.

Kaur, P., Kumar, Y. and Gupta, S., 2022. *Artificial Intelligence Techniques for the Recognition of Multi-Plate Multi-vehicle Tracking Systems: A Systematic Review*. *Archives of Computational Methods in Engineering*, 29(7), pp.4897–4914. https://doi.org/10.1007/s11831-022-09753-4

Khandelwal, R., 2020. *Evaluating performance of an object detection model*. *Towards Data Science* [online]. 6 January. Available at: https://towardsdatascience.com/evaluating-performance-of-an-object-detection-model-137a349c517b [Accessed 5 December 2022].

Khazukov, K., Shepelev, V., Karpeta, T., Shabiev, S., Slobodin, I., Charbadze, I. and Alferova, I., 2020. *Real-time monitoring of traffic parameters*. *Journal of Big Data*, 7(1), article 84. https://doi.org/10.1186/s40537-020-00358-x

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.-Y., Dollár, P. and Girshick, R., 2023. *Segment Anything*. arXiv preprint arXiv:2304.02643. Available at: https://arxiv.org/abs/2304.02643 [Accessed 20 August 2025].

Klinger, N., n.d. *Object Tracking in Computer Vision (Complete Guide)*. *Viso.ai* [online]. Available at: https://viso.ai/deep-learning/object-tracking/ [Accessed 5 December 2022].

Kocur, V., 2019. *Perspective transformation for accurate detection of 3D bounding boxes of vehicles in traffic surveillance*. In: *Proceedings of the 24th Computer Vision Winter Workshop (CVWW)*. Vol. 2, pp.33–41.

Kocur, V. and Ftáčnik, M., 2020. *Detection of 3D bounding boxes of vehicles using perspective transformation for accurate speed measurement*. *Machine Vision and Applications*, 31(7–8), article 62. https://doi.org/10.1007/s00138-020-01117-x

Kocur, V. and Ftáčnik, M., 2021. *Traffic Camera Calibration via Vehicle Vanishing Point Detection*. In: *Artificial Neural Networks and Machine Learning – ICANN 2021*. Lecture Notes in Computer Science, vol. 12895. Cham: Springer, pp.628–639. https://doi.org/10.1007/978-3-030-86383-8_50

Koech, K.E., 2020. On object detection metrics with worked example. *Towards Data Science* [online], 26 August. Available at: https://medium.com/data-science/on-object-detection-metrics-with-worked-example-216f173ed31e [Accessed 5 December 2022].

Koetsier, C., Busch, S. and Sester, M., 2019. *Trajectory extraction for analysis of unsafe driving behaviour*. In: *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W13, pp.1573–1578. https://doi.org/10.5194/isprs-archives-XLII-2-W13-1573-2019

Kolecki, J., Kuras, P., Pastucha, E., Pyka, K. and Sierka, M., 2020. *Calibration of industrial cameras for aerial photogrammetric mapping*. *Remote Sensing*, 12(19), article 3130. https://doi.org/10.3390/rs12193130

Koller, D., Weber, J. and Malik, J., 1994. *Robust multiple car tracking with occlusion reasoning*. In: J.-O. Eklundh, ed. *Computer Vision – ECCV 1994: Proceedings, Lecture Notes in Computer Science*, vol.800. Berlin, Heidelberg: Springer, pp.189–196. https://doi.org/10.1007/3-540-57956-7_22

Krause, J., Stark, M., Deng, J. and Fei-Fei, L., 2013. *3D Object Representations for Fine-Grained Categorization*. In: *2013 IEEE International Conference on Computer Vision Workshops (ICCVW)*. Sydney, Australia: IEEE, pp.554–561. https://doi.org/10.1109/ICCVW.2013.77

Krishna, N., 2022. *Camera Calibration with Example in Python*. *Towards Data Science* [online]. 27 January. Available at: https://medium.com/data-science/camera-calibration-with-example-in-python-5147e945cdeb [Accessed 5 December 2022].

Layek, M.A., Chung, T. and Huh, E.-N., 2019. *Remote distance measurement from a single image by automatic detection and perspective correction*. *KSII Transactions on Internet and Information Systems*, 13(8), pp.3981–4004. https://doi.org/10.3837/tiis.2019.08.009

Lee, Y., Lee, S.-h., Yoo, J. and Kwon, S., 2021. *Efficient single-shot multi-object tracking for vehicles in traffic scenarios*. *Sensors*, 21(19), article 6358. https://doi.org/10.3390/s21196358

Lin, X., Li, C.-T., Sanchez, V. and Maple, C., 2021. *On the detection-to-track association for online multi-object tracking*. arXiv preprint arXiv:2107.00500. Available at: https://arxiv.org/abs/2107.00500 [Accessed 20 August 2025].

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. and Zitnick, C.L., 2014. *Microsoft COCO: Common Objects in Context*. In: *Computer Vision – ECCV 2014. Lecture Notes in Computer Science*, vol. 8693. Cham: Springer, pp.740–755. https://doi.org/10.1007/978-3-319-10602-1_48

Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J. and Zhang, L., 2023. *Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection*. arXiv preprint arXiv:2303.05499. Available at: https://arxiv.org/abs/2303.05499 [Accessed 20 August 2025].

Liu, T. and Liu, Y., 2021. *Deformable model-based vehicle tracking and recognition using 3-D constrained multiple-Kernels and Kalman filter*. *IEEE Access*, 9, pp.90346–90357.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y. and Berg, A.C., 2016. *SSD: Single Shot MultiBox Detector*. In: B. Leibe, J. Matas, N. Sebe and M. Welling, eds. *Computer Vision – ECCV 2016. Lecture Notes in Computer Science*, vol. 9905. Cham: Springer, pp.21–37. https://doi.org/10.1007/978-3-319-46448-0_2

Long, L. and Dongri, S., 2019. *Review of camera calibration algorithms*. In: S.K. Bhatia, S. Tiwari, K.K. Mishra and M.C. Trivedi, eds. *Advances in Computer Communication and Computational Sciences*. Advances in Intelligent Systems and Computing. Singapore: Springer, pp.723–732. https://doi.org/10.1007/978-981-13-6861-5_61

Luvizon, D.C., Nassu, B.T. and Minetto, R., 2016. *A video-based system for vehicle speed measurement in urban roadways*. *IEEE Transactions on Intelligent Transportation Systems*, 18(6), pp.1393–1404. https://doi.org/10.1109/TITS.2016.2606369

Magee, D.R., 2004. *Tracking multiple vehicles using foreground, background and motion models*. *Image and Vision Computing*, 22(2), pp.143–155. https://doi.org/10.1016/S0262-8856(03)00145-8

MathWorks, 2017. *Introduction to Deep Learning: What Are Convolutional Neural Networks?* [online video]. 24 March. Available at: https://www.mathworks.com/videos/introduction-to-deep-learning-what-are-convolutional-neural-networks--1489512765771.html [Accessed 5 December 2022].

Meinhardt, T., Kirillov, A., Leal-Taixé, L. and Feichtenhofer, C., 2021. *TrackFormer: Multi-object Tracking with Transformers. arXiv preprint* arXiv:2101.02702. Available at: https://arxiv.org/abs/2101.02702 [Accessed 20 August 2025].

Miles, V., Gurr, F. and Giani, S., 2022. *Camera-based system for the automatic detection of vehicle axle count and speed using convolutional neural networks*. *International Journal of Intelligent Transportation Systems Research*, 20(3), pp.778–792. https://doi.org/10.1007/s13177-021-00288-4

Minderer, M., Gritsenko, A., Stone, A., Neumann, M., Weissenborn, D., Dosovitskiy, A., Mahendran, A., Arnab, A., Dehghani, M., Shen, Z., Wang, X., Zhai, X., Kipf, T. and Houlsby, N., 2022. *Simple Open-Vocabulary Object Detection with Vision Transformers*. arXiv preprint arXiv:2205.06230. Available at: https://arxiv.org/abs/2205.06230 [Accessed 20 August 2025].

Montgomery, D.C. and Runger, G.C., 2014. *Applied Statistics and Probability for Engineers*. 6th ed. Hoboken, NJ: Wiley. ISBN 9781118539712.

Mukherjee, S. and Das, K., 2013. *An adaptive GMM approach to background subtraction for application in real time surveillance* [online]. arXiv preprint arXiv:1307.5800. Available at: https://arxiv.org/abs/1307.5800 [Accessed 20 August 2025].

Ning, G., Zhang, Z., Huang, C., Ren, X., Wang, H., Cai, C. and He, Z., 2017. Spatially supervised recurrent convolutional neural networks for visual object tracking. In: *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, pp. 1–4. https://doi.org/10.1109/ISCAS.2017.8050867

Orghidan, R., Salvi, J., Gordan, M. and Orza, B., 2012. *Camera calibration using two or three vanishing points*. In: *Proceedings of the 2012 Federated Conference on Computer Science and Information Systems (FedCSIS)*. Wrocław, Poland: IEEE, pp.123–130. Available at: https://annals-csis.org/proceedings/2012/pliks/110.pdf [Accessed 20 August 2025].

Parvin, S., Rozario, L. and Islam, M., 2021. *Vision-Based On-Road Nighttime Vehicle Detection and Tracking Using Taillight and Headlight Features*. *Journal of Computer and Communications*, 9(3), pp.29–53. https://doi.org/10.4236/jcc.2021.93003

Razali, N.M. and Wah, Y.B., 2011. *Power comparisons of Shapiro–Wilk, Kolmogorov–Smirnov, Lilliefors and Anderson–Darling tests*. *Journal of Statistical Modeling and Analytics*, 2(1), pp.21–33.

Reinsch, C.H., 1967. *Smoothing by spline functions*. *Numerische Mathematik*, 10, pp.177–183. https://doi.org/10.1007/BF02162161

Ren, S., He, K., Girshick, R. and Sun, J., 2015. *Faster R-CNN: Towards real-time object detection with region proposal networks*. *Advances in neural information processing systems*, *28*. arXiv preprint arXiv:1506.01497. Available at: https://arxiv.org/abs/1506.01497 [Accessed 21 August 2025].

Ridder, C., Munkelt, O. and Kirchner, H., 1995. *Adaptive background estimation and foreground detection using Kalman-filtering*. In: *Proceedings of the International Conference on Recent Advances in Mechatronics (ICRAM '95)*, Istanbul, Turkey, 14–16 Aug 1995, pp. 193–199. Available at: https://citeseerx.ist.psu.edu/document?doi=f680e7e609ce0729c8a594336e0cf8f447b3ef13&repid=rep1&type=pdf [Accessed 21 August 2025].

Rosebrock, A., 2022. *YOLOv7 object detection paper explanation and inference*. LearnOpenCV [online]. Available at: https://learnopencv.com/yolov7-object-detection-paper-explanation-and-inference/ [Accessed 21 August 2025]

Savitzky, A. and Golay, M.J.E., 1964. *'Smoothing and differentiation of data by simplified least squares procedures'*, *Analytical Chemistry*, 36(8), pp. 1627–1639. Available at: https://doi.org/10.1021/ac60214a047

Shapiro, S.S. and Wilk, M.B., 1965. *An analysis of variance test for normality (complete samples)*. *Biometrika*, 52(3–4), pp. 591–611. Available at: https://doi.org/10.1093/biomet/52.3-4.591

Shashirangana, J., Padmasiri, H., Meedeniya, D. and Perera, C., 2020. *Automated license plate recognition: A survey on methods and techniques*. *IEEE Access*, 9, pp. 11203–11225. Available at: https://doi.org/10.1109/ACCESS.2020.3047929

Shumway, R.H. and Stoffer, D.S., 2017. *Time series analysis and its applications: with R examples*. 4th ed. Cham: Springer. Available at: https://doi.org/10.1007/978-3-319-52452-8

Schirra, J.R.J., Bosch, G., Sung, C.K. and Zimmermann, G., 1987. *From image sequences to natural language: A first step toward automatic perception and description of motions*. *Applied Artificial Intelligence*, 1(4), pp. 287–305. Available at: https://doi.org/10.1080/08839518708927976

Smoothing Images. *OpenCV – Open Source Computer Vision* [online]. Available at: https://docs.opencv.org/4.x/d4/d13/tutorial_py_filtering.html [Accessed 5 December 2022].

Sochor, J., 2018. *Automatic traffic video surveillance: fine-grained recognition of vehicles and automatic speed measurement*. Dissertation thesis. Brno: University of Technology Brno.

Sochor, J., Juránek, R. and Herout, A., 2017. *Traffic surveillance camera calibration by 3D model bounding box alignment for accurate vehicle speed measurement*. *Computer Vision and Image Understanding*, 161, pp. 87–98. Available at: https://doi.org/10.1016/j.cviu.2017.05.015

Sochor, J., Juránek, R., Špaňhel, J., Maršík, L., Široký, A., Herout, A. and Zemčík, P., 2017. *BrnoCompSpeed: review of traffic camera calibration and comprehensive dataset for monocular speed measurement*. *arXiv preprint*, arXiv:1702.06441. Available at: https://arxiv.org/abs/1702.06441 [Accessed 21 August 2025].

Sochor, J., Juránek, R., Špaňhel, J., Maršík, L., Široký, A., Herout, A. and Zemčík, P., 2018. Comprehensive data set for automatic single camera visual speed measurement. *IEEE Transactions on Intelligent Transportation Systems*, 20(5), pp. 1633–1643. Available at: https://doi.org/10.48550/arXiv.1702.06441

Song, H., Liang, H., Li, H., Dai, Z. and Yun, X., 2019. *Vision-based vehicle detection and counting system using deep learning in highway scenes*. *European Transport Research Review*, 11(1), 51. Available at: https://doi.org/10.1186/s12544-019-0390-4

Song, J., Fan, Y., Song, H. and Zhao, H., 2022. *Target tracking and 3D trajectory reconstruction based on multicamera calibration. Journal of Advanced Transportation*, Article ID 5006347. doi:10.1155/2022/5006347.

Song, Y.M., Noh, S.J., Yu, J., Park, C.-W. and Lee, B.-G., 2014. Background subtraction based on Gaussian mixture models using color and depth information. In: *2014 International Conference on Control, Automation and Information Sciences (ICCAIS)*. IEEE, pp. 132–135. Available at: https://doi.org/10.1109/ICCAIS.2014.7020544.

Stáňa, I., 2016. *Technický výklad vybraných ustanovení zákona č. 361/2000 Sb., o provozu na pozemních komunikacích* [Master's thesis]. Brno: Brno University of Technology, Institute of Forensic Engineering.

Tang, X., Song, H., Wang, W. and Yang, Y., 2020. Vehicle spatial distribution and 3D trajectory extraction algorithm in a cross-camera traffic scene. *Sensors*, 20(22), 6517. Available at: https://doi.org/10.3390/s20226517

Tang, Z., Lin, Y.S., Lee, K.H., Hwang, J.N. and Chuang, J.H., 2019. *ESTHER: joint camera self-calibration and automatic radial distortion correction from tracking of walking humans*. *IEEE Access*, 7, pp. 10754–10766.

Tang, Z., Naphade, M., Liu, M.Y., Yang, X., Birchfield, S., Wang, S., et al., 2019. CityFlow: a city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2019)*, pp. 8797–8806. Available at: https://doi.org/10.1109/CVPR.2019.00900

TensorFlow Tutorials. *TensorFlow* [online], 2022. Available at: https://www.tensorflow.org/tutorials [Accessed 5 December 2022].

Thode, H.C., 2002. *Testing for Normality.* New York: Marcel Dekker.

Tokař, S., 2014. *Analýza pojmů náhle x nenáhle pomocí experimentu*. In: Schejbal, J. and Bradáč, A. (eds.) *Sborník příspěvků konference Expert Forensic Science 2014*. Brno: Vysoké učení technické v Brně, Ústav soudního inženýrství, pp. 283–295. ISBN 978-80-214-4852-0.

Two sample t-test and z-test. *XLSTAT* [online]. Available at: https://www.xlstat.com/en/solutions/features/two-sample-t-and-z-tests [Accessed 6 December 2022].

Uijlings, J.R., Van De Sande, K.E., Gevers, T. and Smeulders, A.W., 2013. *Selective search for object recognition*. *International Journal of Computer Vision*, 104(2), pp. 154–171. Available at: https://doi.org/10.1007/s11263-013-0620-5.

United Nations Economic Commission for Europe (UNECE) (latest revision) *UN/ECE Regulation No. 13-H: Uniform provisions concerning the approval of passenger cars with regard to braking.* Geneva: UNECE.

Vasconcelos, F., Barreto, J.P. and Boyer, E., 2017. *Automatic camera calibration using multiple sets of pairwise correspondences*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4), pp. 791–803. Available at: https://doi.org/10.1109/TPAMI.2017.2699648

Voulodimos, A., Doulamis, N., Doulamis, A. and Protopapadakis, E., 2018. *Deep learning for computer vision: a brief review*. *Computational Intelligence and Neuroscience*, 2018, Article ID 7068349. Available at: https://doi.org/10.1155/2018/7068349

Wang, C.Y., Bochkovskiy, A. and Liao, H.Y.M., 2022. *YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*. *arXiv preprint*, arXiv:2207.02696. Available at: https://arxiv.org/abs/2207.02696 [Accessed 21 August 2025].

Weng, X., Wang, J., Levine, S. and Kitani, K., 2020. *AB3DMOT: a baseline for 3D multi-object tracking and new evaluation metrics*. *arXiv preprint*, arXiv:2008.08063. Available at: https://doi.org/10.48550/arXiv.2008.08063

Wilk, M.B. and Gnanadesikan, R., 1968. Probability plotting methods for the analysis of data. *Biometrika*, 55(1), pp. 1–17. Available at: https://doi.org/10.1093/biomet/55.1.1

Wojke, N., Bewley, A. and Paulus, D., 2017. *Simple online and realtime tracking with a deep association metric*. In: *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 3645–3649. IEEE. Available at: https://doi.org/10.1109/ICIP.2017.8296962

Wu, F., Song, H., Dai, Z., Wang, W. and Li, J., 2021. Multi-camera traffic scene mosaic based on camera calibration. *IET Computer Vision*, 15(1), pp. 47–59. Available at: https://doi.org/10.1049/cvi2.12009

Xu, H. and Wang, X., 2012. *Camera calibration based on perspective geometry and its application in LDWS*. *Physics Procedia*, 33, pp. 1626–1633. Available at: https://doi.org/10.1016/j.phpro.2012.05.262

Aharon, N., Orfaig, R. and Bobrovsky, B.-Z., 2022. *BoT-SORT: Robust Associations Multi-Pedestrian Tracking. arXiv preprint*, arXiv:2206.14651. doi:10.48550/arXiv.2206.14651.

Yaghoobi Ershadi, N., Menéndez, J.M. and Jiménez, D., 2018. *Robust vehicle detection in different weather conditions: using MIPM*. *PLOS ONE*, 13(3), e0191355. Available at: https://doi.org/10.1371/journal.pone.0191355

Yang, Y. and Bilodeau, G.-A., 2017. *Multiple object tracking with kernelized correlation filters in urban mixed traffic*. In: *14th Conference on Computer and Robot Vision (CRV 2017)*, pp. 209–216. Available at: https://doi.org/10.1109/CRV.2017.18

Yao, L., Zhao, Y., Fan, J., Liu, M., Jiang, J. and Wan, Y., 2019. *Research and application of license plate recognition technology based on deep learning*. *Journal of Physics: Conference Series*, 1237(2), p. 022155. IOP Publishing. Available at: https://doi.org/10.1088/1742-6596/1237/2/022155

Yu, Z., Jiang, Q. and Li, X., 2020. *MPP: a novel algorithm for estimating vehicle space headways from a single image*. *Journal of Advanced Transportation*, 2020, Article ID 5715686. Available at: https://doi.org/10.1155/2020/5715686

Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L.M. and Shum, H.-Y., 2022. *DINO: detr with improved denoising anchor boxes for end-to-end object detection*. In: *11th International Conference on Learning Representations (ICLR 2023)*. *arXiv preprint*, arXiv:2203.03605. Available at: https://arxiv.org/abs/2203.03605 [Accessed 21 August 2025].

Zhang, W., Song, H., Liu, L., Li, C., Mu, B. and Gao, Q., 2021. *Vehicle localisation and deep model for automatic calibration of monocular camera in expressway scenes*. *IET Intelligent Transport Systems*, 16(4), pp. 459–473. Available at: https://doi.org/10.1049/itr2.12152

Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W. and Wang, X., 2022. *ByteTrack: multi-object tracking by associating every detection box*. In: *Computer Vision – ECCV 2022*, Lecture Notes in Computer Science, vol. 13668, pp. 1–21. Springer, Cham. Available at: https://link.springer.com/chapter/10.1007/978-3-031-20047-2_1

Zhang, Y., Wang, C., Wang, X., Zeng, W. and Liu, W., 2021. *FairMOT: on the fairness of detection and re-identification in multiple object tracking*. *International Journal of Computer Vision*, 129(11), pp. 3069–3087. Available at: https://doi.org/10.1007/s11263-021-01513-4

Zhang, Y., Zhao, C., He, J. and Chen, A., 2016. *Vehicles detection in complex urban traffic scenes using Gaussian mixture model with confidence measurement*. *IET Intelligent Transport Systems*, 10(6), pp. 416–423. Available at: https://doi.org/10.1049/iet-its.2015.0141

Zhang, Z., 2000. *A flexible new technique for camera calibration*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11), pp. 1330–1334. Available at: https://doi.org/10.1109/34.888718

Zhao, M., Zhong, Y., Sun, D. and Chen, Y., 2021. *Accurate and efficient vehicle detection framework based on SSD algorithm*. *IET Image Processing*, 15(13), pp. 3094–3104. Available at: https://doi.org/10.1049/ipr2.12297

Zheng, L., Shen, L., Tian, L., Wang, S., Wang, J. and Tian, Q., 2015. *Scalable person re-identification: a benchmark*. In: *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1116–1124. IEEE. Available at: https://doi.org/10.1109/ICCV.2015.133

Zhou, X., Koltun, V. and Krähenbühl, P., 2020. *Tracking objects as points*. In: *Computer Vision – ECCV 2020*, Lecture Notes in Computer Science, vol. 12347, pp. 474–490. Springer, Cham. Available at: https://doi.org/10.1007/978-3-030-58548-8_28

Zhu, M., Zhang, S., Zhong, Y., Lu, P., Peng, H. and Lenneman, J., 2021. *Monocular 3D vehicle detection using uncalibrated traffic cameras through homography*. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3814–3821. IEEE. Available at: https://doi.org/10.1109/IROS51168.2021.9636384

Zhu, X., Su, W., Lu, L., Li, B., Wang, X. and Dai, J., 2020. *Deformable DETR: deformable transformers for end-to-end object detection*. *arXiv preprint*, arXiv:2010.04159. Available at: https://arxiv.org/abs/2010.04159 [Accessed 21 August 2025].

Zivkovic, Z., 2004. *Improved adaptive Gaussian mixture model for background subtraction*. In: *Proceedings of the 17th IEEE International Conference on Pattern Recognition (ICPR 2004)*, Vol. 2, pp. 28–31. Available at: https://doi.org/10.1109/ICPR.2004.1333992

Zivkovic, Z. and van der Heijden, F., 2006. *Efficient adaptive density estimation per image pixel for the task of background subtraction*. *Pattern Recognition Letters*, 27(7), pp. 773–780. Available at: https://doi.org/10.1016/j.patrec.2005.11.005

Zwemer, M., Scholte, D., Wijnhoven, R. and de With, P.H.N., 2022. *3D detection of vehicles from 2D images in traffic surveillance*. In: *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2022)*, Volume 5 – VISAPP, pp. 97–106. SciTePress Digital Library. Available at: https://doi.org/10.5220/0010783600003124

8. AUTHOR'S PUBLICATIONS

Jilek, P., Šefčík, I., Verner, J. and Berg, J., 2019. *System allowing adhesion force change of road vehicle. Engineering for Rural Development: Proceedings of the 18th International Scientific Conference Engineering for Rural Development (ERD 2019)*, Jelgava, Latvia, 22–24 May. Jelgava: Latvia University of Agriculture, pp. 1876–1882. ISSN 1691-3043. e-ISSN 1691-5976.

Jilek, P., Krmela, J. and Berg, J., 2020. *Modification of the adhesive force on a vehicle by reducing the radial force reaction of the wheels. Transport Problems 2020: Proceedings of the XII International Scientific Conference and IX International Symposium of Young Researchers (TRANSPORT PROBLEMS 2020)*, Katowice, Poland, 30 November–2 December. Katowice: Silesian University of Technology, pp. 345–353. ISBN 978-83-959742-0-5. ISSN 1896-0596.

Jilek, P., Krmela, J. and Berg, J., 2021. *Modification of the adhesive force by changing the radial reaction on vehicle wheels. Transport Problems*, 16(1), pp. 179–186. doi:10.21307/tp-2021-015. ISSN 1896-0596. e-ISSN 2300-861X.

Jilek, P. and Berg, J., 2021. *Optimization of device allowing variation of adhesion force for road vehicle testing at safe speed. Engineering for Rural Development*, 20, pp. 373–378. Jelgava: Latvia University of Life Sciences and Technologies. doi:10.22616/ERDev.2021.20.TF078. ISSN 1691-3043. e-ISSN 1691-5976.

Jilek, P. and Berg, J., 2021. *Umístění svaru a jeho vliv na koncentraci napětí. Perner's Contacts*, 16(2). doi:10.46585/pc.2021.2.1738. ISSN 1801-674X.

Jilek, P. and Berg, J., 2021. *The adhesion force change of an experimental road vehicle. Manufacturing Technology*, 21(5), pp. 634–639. doi:10.21062/mft.2021.083. ISSN 1213-2489.

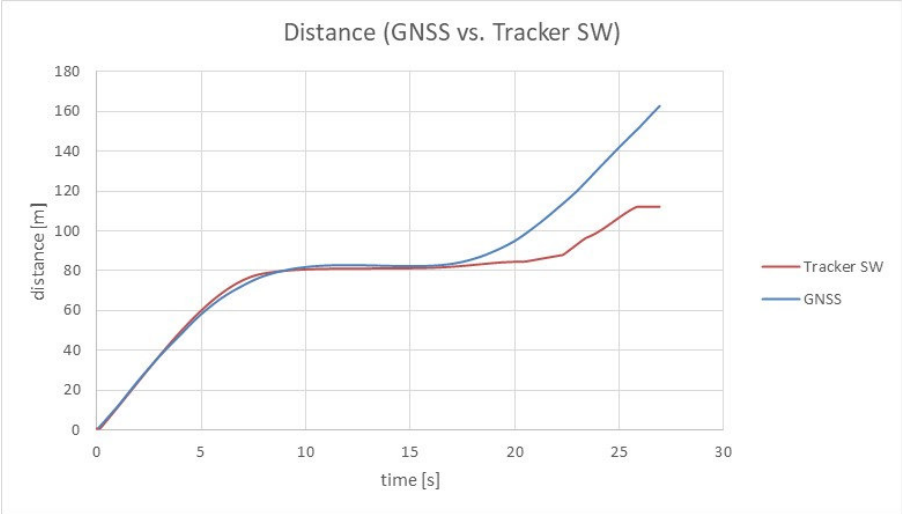
Bradáč, A., Semela, M., Kledus, R., Bilík, M., Křižák, M., Mikulec, R., Bucsuházy, K., Belák, M., Mičunek, T., Frydrýn, M., Nouzovský, L., Svatý, Z., Radová, Z., Fryšták, M., Vojtíšek, T., Pokorný, J. and Berg, J., 2021. *Teorie a praxe analýzy silničních nehod*. Brno: VUTIUM. ISBN 978-80-214-6209-0. doi:10.13164/usi.book.tpasn.

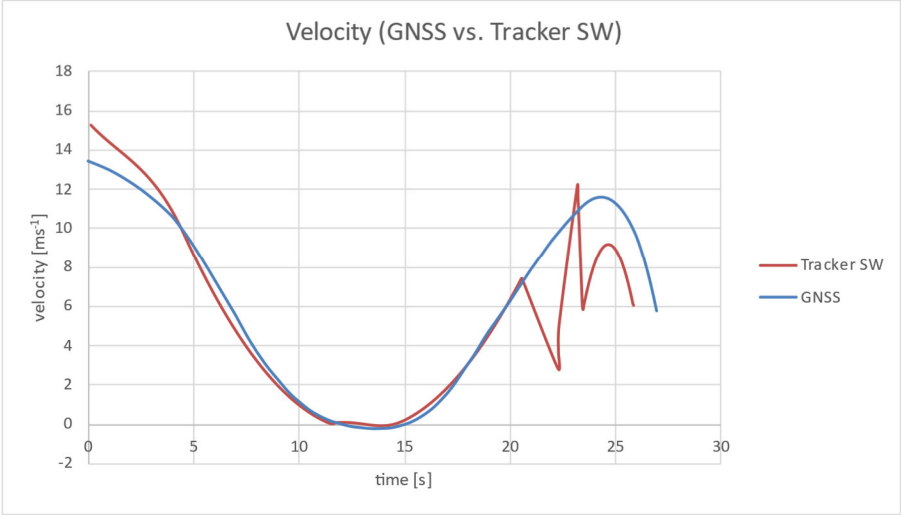
Berg, J., Jilek, P., Pokorný, J. and Krmela, J., 2022. *Metody předzpracování obrazu pro automatickou detekci účastníků silničního provozu. Perner's Contacts*, 17(2). doi:10.46585/pc.2022.2.2389. ISSN 1801-674X.

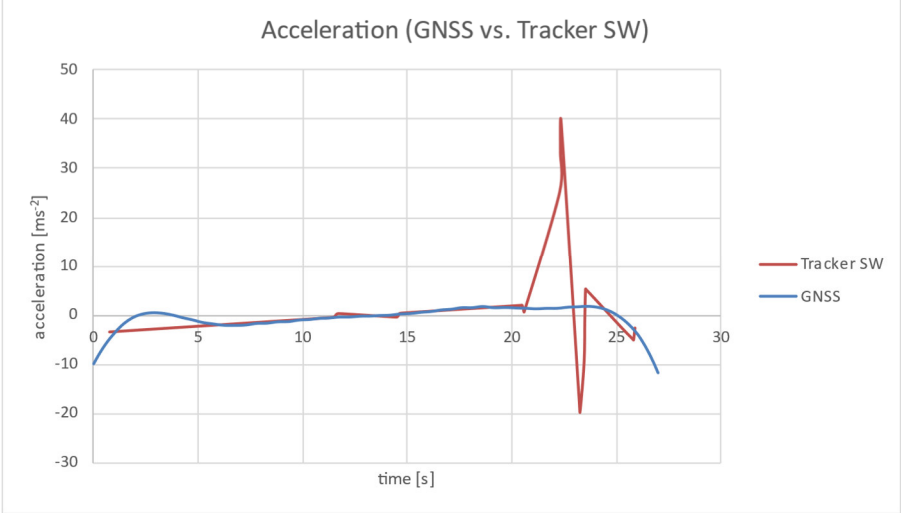
Jilek, P., Berg, J. and Sadjiep Tchuigwa, B.S., 2022. *Influence of the weld joint position on the mechanical stress concentration in the construction of the Alternative Skid Car system's skid chassis. Applied Sciences – Basel*, 12(1), p. 397. doi:10.3390/app12010397. ISSN 2076-3417.

APPENDIX 1: SCENE PHYSICAL VALIDATION

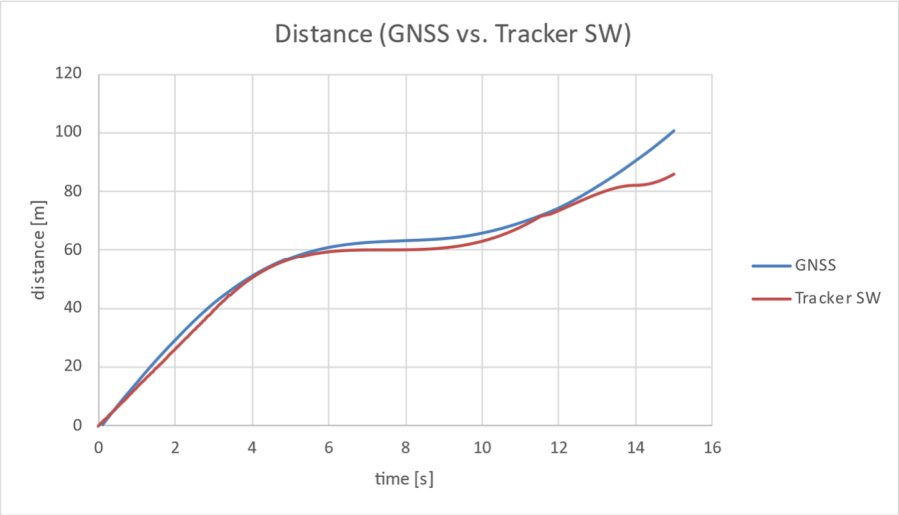
Validation of the measurement number 1 between data from GNSS station and tracker SW.

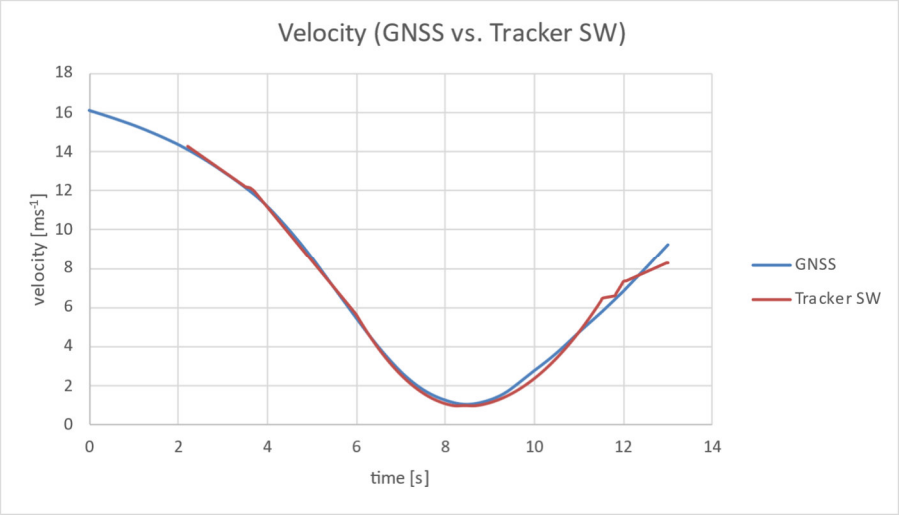


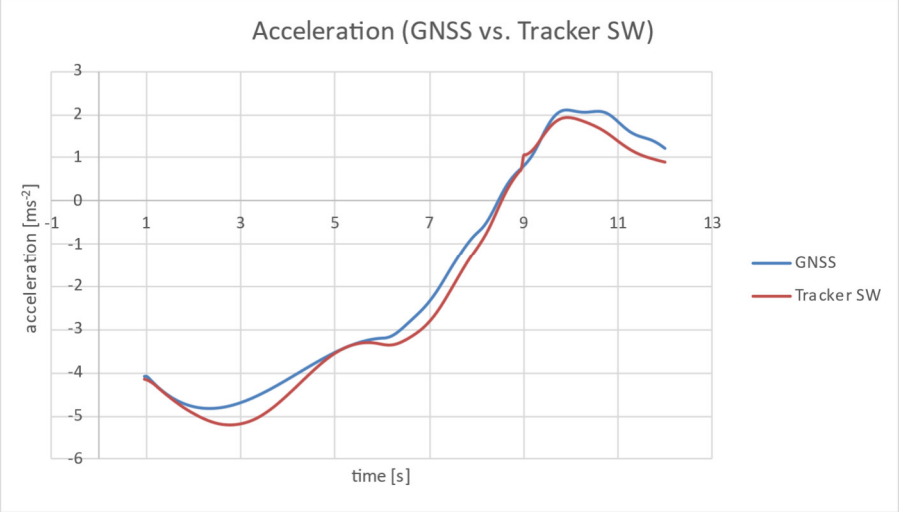




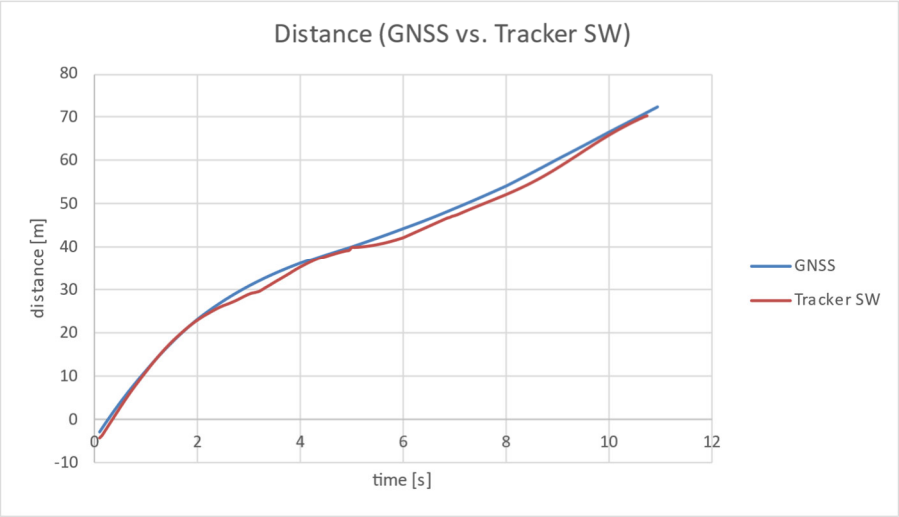
Validation of the measurement number 2 between data from GNSS station and tracker SW.

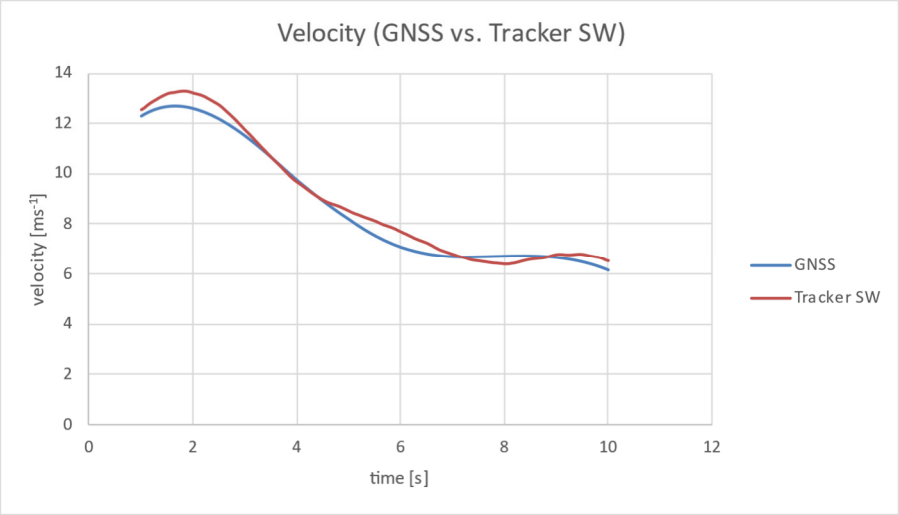


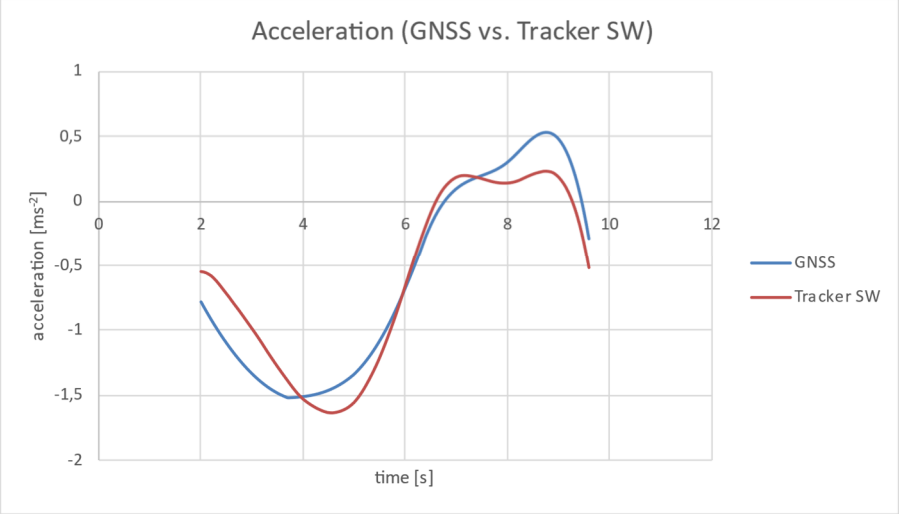




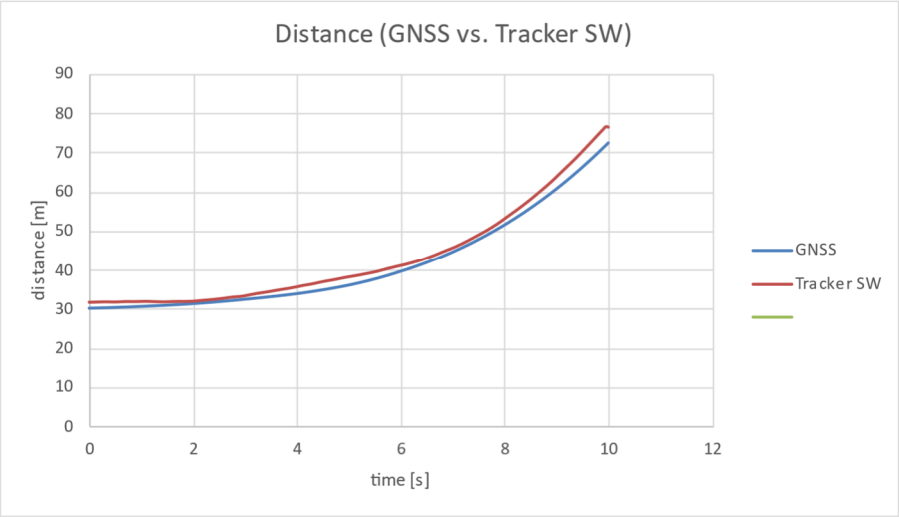
Validation of the measurement number 3 between data from GNSS station and tracker SW.

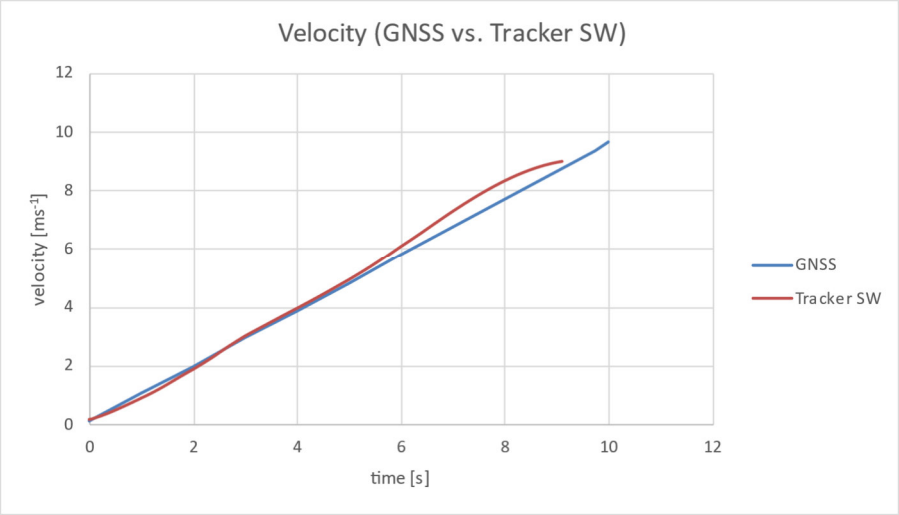


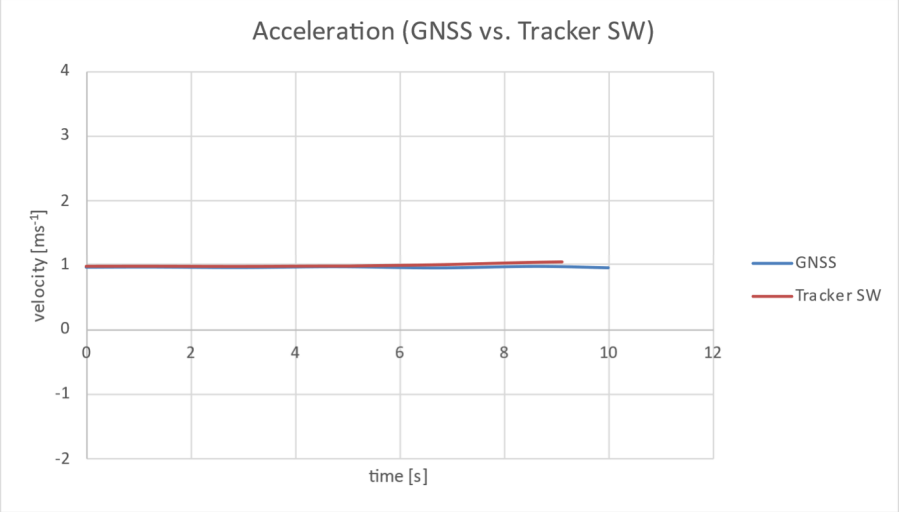




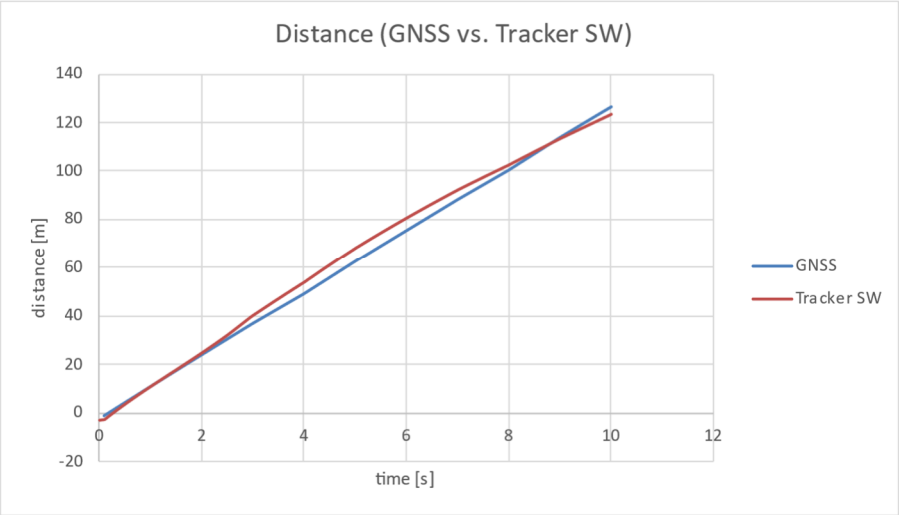
Validation of the measurement number 4 between data from GNSS station and tracker SW.

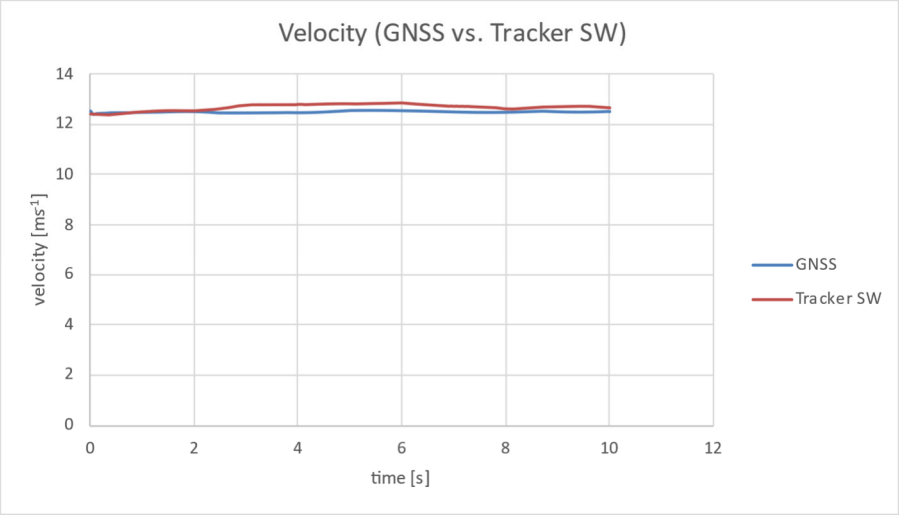


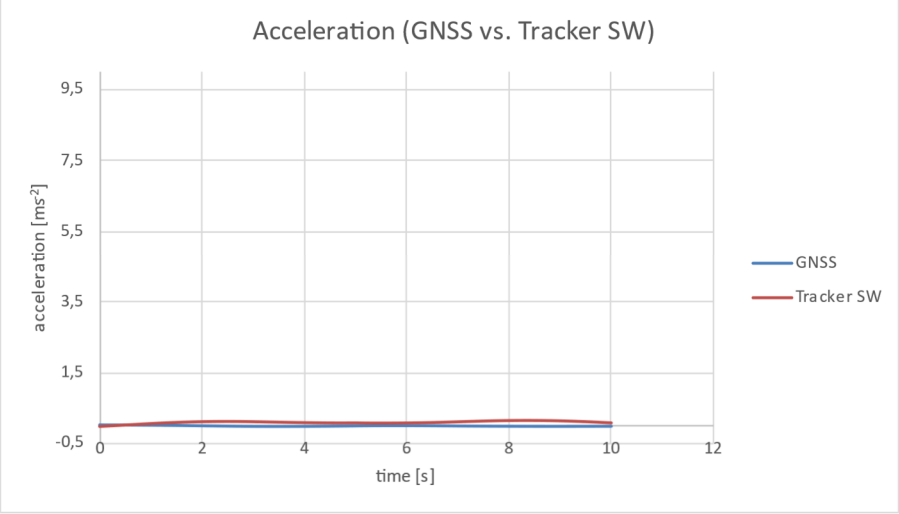




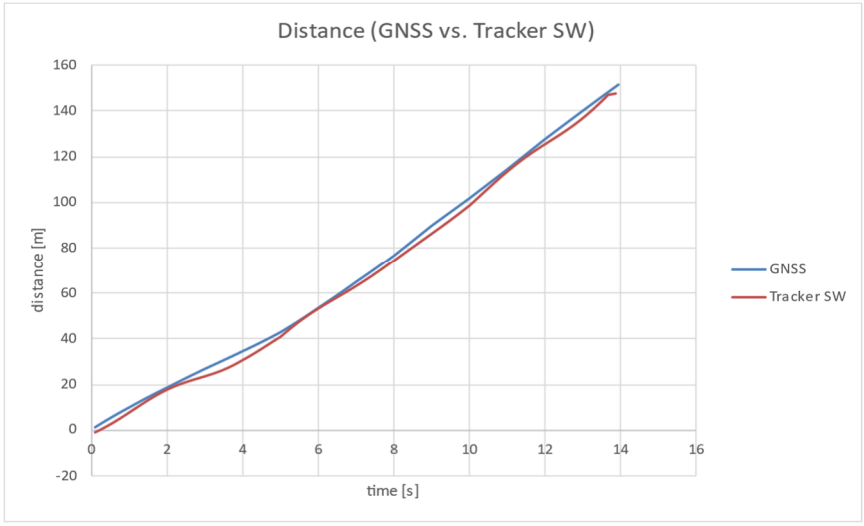
Validation of the measurement number 5 between data from GNSS station and tracker SW.

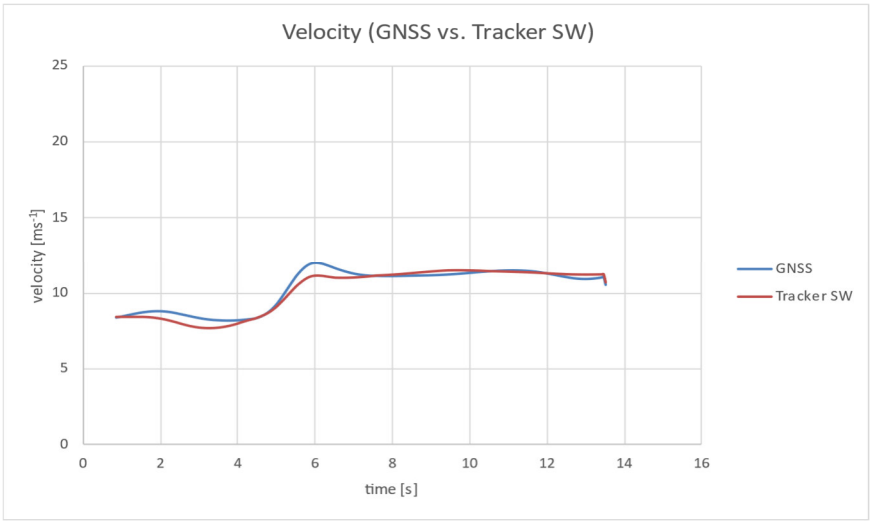


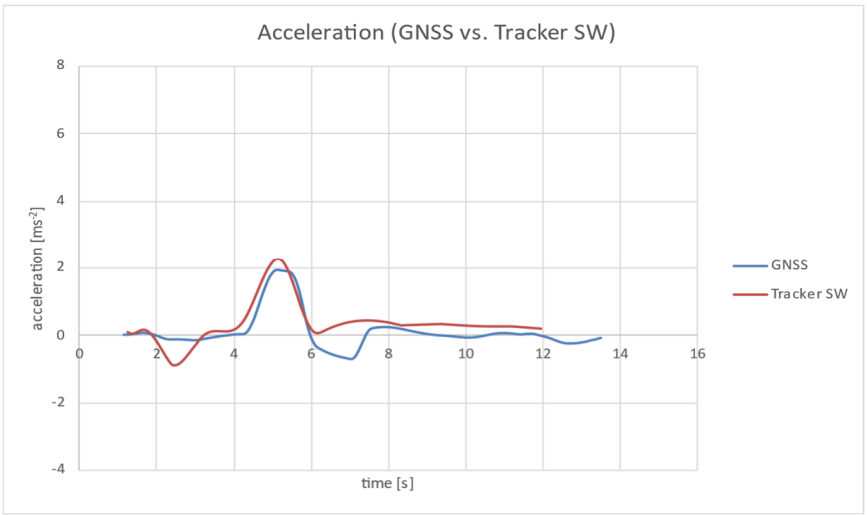




Validation of the measurement number 6 between data from GNSS station and tracker SW.







APPENDIX 2: SW ARCHITECTURE BLOCK DIAGRAM

In this chapter, an architecture of the detection and tracking SW is visualized. In the first Figure, a high-level structure is displayed, in the other Figure, a more detailed diagram of the detector and tracker blocks is provided.