Technology Report
PoC GLAT
Supporting Technologies
VIDEO / FOS

2020

smartrail 4.0: Technology Report PoC GLAT – Video / FOS

Status	Released
Version	Version 01-01
Last Change	10. Juni 2020
Copyright	This document is protected by copyright. Any commercial use requires prior, explicit permission.
File:	www.smartrail40.ch

Table of contents

1 Preamble

This present technology report Video / FOS describes the status of all activities regarding the supporting localisation technologies Video / FOS of the last 16 months which have been triggered by the commissioning of the GLAT technology PoC on 6 June 2018.

After the publication of the interim report (Zwischenbericht PoC) [1] beginning of 2019, it was decided at the request of the GLAT project management to focus the activities on the sensor technologies: GNSS, IMU and wheel odometry as well as their fusion, which resulted in a technology report (Technologiebericht PoC) [2] on these topics which was completed by the end of 2019. In addition, a separate technology report for the supporting technologies FOS and video, has been commissioned for April 2020, which resulted in the present document.

Compared to the interim report [1], many of the qualitative statements in the technology report are now quantitatively evaluated and presented on the basis of measurements. The report focuses on the question of whether and under what conditions the technical and architectural feasibility of the GLAT system is given.

Target audience

The technology report addresses the core team smartrail 4.0, the Federal Office of Transport FOT and all interested smartrail 4.0 partners. It will be presented to the core team and published on https://www.smartrail40.ch.

1.1 Authors Technology Report

Name	Company	Project Function	Chapters involved / Role
Dr. Christian Robl	M2C ExpertControl GmbH	Project Manager	2,3,4,5,6 / Author
Dr. Eduardo Rubino	M2C ExpertControl GmbH	Expert FOS / Video	4,6 / Author
Dr. Daniele Capriotti	M2C ExpertControl GmbH	Expert Video	3, 6 / Author
Prof. Dr.-Ing. Darius Burschka	TUM – Technical University Munich	Department of Informatics, Telerobotics and Sensor Data Fusion	Reviewer
Marko Pavlic	M2C ExpertControl GmbH	Specialist Control Design and Signal Processing FOS	3,6 / Author
Angelika Rettinger	M2C ExpertControl GmbH	Strategy & Business Development	2, Reviewer/Author
Dr. Alasdair Murray	Optasense	Signal Analyst	Reviewer
Andrew Hall	Optasense	Signal Analyst	Reviewer
Kevin Molloy	Optasense	Project Manager – Transport Systems	Reviewer
Lothar Jöckel	SBB Platform for Research and Innovation (PFI)	IT-/ Video Specialist	2,7 / Author
Dr. Albert Hofstetter	SBB Platform for Research and Innovation (PFI)	Senior Research Scientist	2,7 / Author

Dr. Alex Brand	SBB SmartRail 4.0	Program Manager LCS	Reviewer
Urs Ackermann	SBB SmartRail 4.0	Project Manager GLAT and AWAP	Reviewer
Sebastian Ohrendorf-Weiss	SBB SmartRail 4.0	Project Manager GLAT	Reviewer
Thomas Düsel	accellence GmbH	Project Manager Measurement Se-ries	1,2 Author and Overall coordinator Technology Report

Table 1-1 Authors Technology Report

2 General view Technology PoC GLAT – Video / FOS

2.1 Aims and objectives

The aim is a proof of concept of the supporting localisation technologies Video and Fiber Optic Sensing (FOS) answering as much as possible the following fundamental questions:

- To what extent can these technologies be used for reliable and accurate localisation in the railway environment?
- Under which conditions does the respective technology work?
- What are the options and restrictions regarding possible applications?
- What level of accuracy and availability can be achieved in the measurement runs?
- Can qualitative and quantitative statements be made regarding the determination of accuracy?
- Which prerequisites must be met?

The technology PoC shall pursue a more detailed investigation on the sensor technologies Video and Fiber Optic Sensing (FOS) for accurate and reliable train localisation and, if possible, applicable on the entire network of railways in Switzerland.

For optimisation and testing, single measurement runs shall be carried out with the various sensor technologies. To compare the different technologies, a run with all sensors installed in parallel shall be performed.

The results shall be checked against ground truth, e.g. axle counters, balises and/or GNSS/IMU data. The potential and the capability of the individual technologies shall be accordingly assessed. Another important aspect is certifiability of the approaches. This is essential for deploying localisation in railways for dedicated use cases and therefore it shall be considered carefully (see 5.2).

Derived from the overall objectives the technology-specific fields of action and objectives are listed in the following sections:

Sensor Technology Video

The following aims and objectives shall be taken into account for the sensor technology Video:

- Improving camera setup for railway application
  - with main focus on ease of use and reliable object detection
- Railway identification and camera calibration
  - Automatic track detection
  - Automatic extrinsic camera calibration
- Train localisation by Visual Odometry
  - Relative localisation of train
    - calculation of relative train position and distance travelled
    - reduction of drift
    - estimation of the confidence of the results (error distribution)
  - Absolute localisation of train
    - using landmarks for determining the absolute train position and correcting errors and uncertainties
- Object recognition
  - automatic detection of infrastructure objects, e.g. stopping plates, AprilTags, points, by using the camera system
- Realtime capable and deterministic algorithms with respect to certifiability
- Investigations on robustness under various weather and light conditions
- Proving results by measurement run by comparing to GNSS / IMU and Track Topography (GTG)

Sensor Technology FOS

The following aims and objectives shall be taken into account for the sensor technology FOS:

- Calibration of the fiber cable with respect to the track
- Tracking of
    - moving trains
    - their position
    - their front and rear ends

  at different velocities
- Determining the
    - length of train, i.e. train integrity
    - velocity
- Estimation of the confidence of the result (error distribution)
- Realtime capable and deterministic algorithms with respect to certifiability
- Investigations on robustness under various weather conditions and by given disturbances such as traffic on a nearby motorway
- Proving results by measurement run with measurement train and applying FOS on all standard trains travelling in a given time slot

SBB Innovation project Optical Train Localisation

In the SBB Innovation project “Optical Train Localisation” we investigate an initial proof of concept (PoC), for a deep learning based optical approach for exact train localisation. The presented PoC is performed in three iterations.

In the first iteration, we investigate the following objectives:

- Optical detection and recognition of tracks and selection of track that was driven on
- Optical detection of further objects of interest along the tracks

In the second iteration, we investigate the following objectives:

- Integration of topology database (DfA) with optical track selection to obtain a track specific localisation.
- Investigation of the robustness of the optical detection with respect to further lighting and weather conditions as well as for further routes.

In the third iteration, we investigate the following objectives:

- Optical detection and recognition of mast boards to determine the longitudinal position of the train
- Optical detection and recognition of kilometre panels to determine the longitudinal position of the train

  Optical detection of switch state and expected driveway of the train

2.2 Initial Situation (Technologiebericht-PoC_GLAT_v1.00 [1])

Localisation is essential for safe rail systems. Nowadays, absolute train localisation is based on infrastructure and only at particular checkpoints, e.g. using balises. A continuous localisation in real-time is the next step in rail automation. In the following, different solutions and concepts are considered and assessed.

This report mainly focuses on the sensor technologies Video and Fiber Optic Sensing (FOS) including deterministic and certifiable approaches of image and signal processing algorithms. In addition, the SBB innovation project “Optical Train Localisation” features a deep-learning approach on the available data, which is non-deterministic.

Sensor Technology Video

Previous investigations showed huge potentials for Visual Odometry and Video Localisation for relative and absolute train localisation. Both technologies basically work under good weather and light conditions, but reliability, availability and accuracy shall be improved and assessed, also under non-optimal conditions.

Basically, Video is divided into Visual Odometry for local or relative localisation and into Video Localisation for absolute localisation, as they were also considered in [1] and [2]. The previous report [1] proves the potentials and challenges of these technologies which may play a significant role in train localisation in the future.

The main benefit of Visual Odometry is the possibility for a slip-free odometry. Video Localisation provides absolute train positions and may replace balises in the future. Furthermore, it can be used to support Visual Odometry, e.g. for increasing accuracy over a long distance.

In a previous measurement run between Thun and Burgdorf, 3 TByte of video data were recorded under varying light conditions and at different train speeds. Two algorithm approaches were evaluated, i.e. speed calculation by optical flow of the video images and by optical mouse tracking. It could be shown that Visual Odometry basically works under good weather and light conditions.

However, it is highly important to choose and calibrate an appropriate camera system. In this context, a stereo camera has no benefit compared to a mono-camera and an illumination could improve the quality of the video images significantly. The control and calibration of the camera and its settings need to be optimally adapted to the problem and the camera installation needs to be improved in order to suppress reflections and pitching movements. In addition, the reliability of the algorithms shall be improved and verified with respect to the objectives mentioned above. All of this is scope of this report.

Regarding Video Localisation another previous measurement run from Münsingen to Uttingen showed, that the camera system in use was able to detect all larger reference points (AprilTags [3], 64x64 cm), but it could not recognize the smaller ones (16x16 cm) for absolute train localisation at lower and higher velocities.

Therefore, it is important to use tags of the right size and to mount them on appropriate spots, while the right size depends on the calibration of the camera system chosen for Video Localisation. It would even be better to use existing infrastructure elements with an exactly known position as reference points. Again, the algorithms and the control of the camera shall be improved regarding availability and reliability and further validations will be needed according to the objectives mentioned in chapter 2.1.

Sensor Technology Fiber Optic Sensing (FOS)

FOS is a track-side technology for absolute localisation and determination of train position. It can be used as a complement method, especially for sections with limited availability of GNSS or mobile communications. One advantage of this technology is that the fiber cables are already installed for data transfer and communications. Recent investigations showed various challenges, e.g. regarding accuracy, acoustic disturbances, low trains speeds etc. This report presents various approaches and introduces real-time analysis.

Measurement runs have already been made and FOS data is available. The initial situation is that the quality of the localisation needs further investigation and that the localisation is not satisfiable at low train speeds, e.g. below 40 km/h. More measurements and long-term testing are needed for validation of FOS.

Also, the initial report [1] did not take into consideration the current real time requirements and was able to use tools for post analysis which are not available in real time. It was a proof-of-concept which analysed the whole interval of time as one block and was only run with a single set of data. The current report expands greatly in relation to the first one and also employs real time analysis tools which do all the necessary processing in real time. It also introduces new measures which are able to deal with the varying attenuation of each fiber channel and also have the potential for higher accuracy than simply using the power of the signal.

At the moment there are investigations how these technologies can contribute to an accurate and safe localisation.

SBB Innovation Project optical train localisation

Deep-learning algorithms, especially based on convolutional neural networks (CNNs), have led to a huge improvement in many computer vision areas, such as object detection and distance estimation. For many applications, from cancer detection to self-driving cars, object detection based on CNNs has already been investigated and holds great promises. However, the suitability of deep learning-based approaches for optical train localisation has not yet been investigated. Further, the process to reach a SIL4 certification, as required for train localisation, is not yet well established for machine learning based approaches. As a first step, this report investigates the usefulness and the reliability of the machine learning algorithms for optical train localisation in a first proof of concept.

2.3 Results and findings

The following sections provide a summary of the main results and key findings of the relevant localisation approaches. Detailed results and findings are described in the corresponding chapters referred to in the text below.

In all cases, time synchronisation among the sensors is crucial for their integration and there should be a high-quality reference clock for all of them. Furthermore, a valid and accurate ground truth is essential to assess the performance of the sensors.

Sensor Technology Video

The results reveal a huge potential for Visual Odometry and Video Localisation as part of a future continuous, safe and accurate train localisation. However, the excellent results obtained still have some room for improvement.

Extent of use in railways:

- Visual Odometry has a high precision for relative localisation; a combination with other sensor technologies, e.g. GNSS, seems very promising.
- Video Localisation can be used standalone for absolute localisation for dedicated use cases.
- Virtual Balises could be introduced, e.g. using infrastructure objects like point-frogs as a global reference for the generation of TPRs within ETCS.
- Compact camera system setup incl. autocalibration allowing for easy installation and train localisation with minimal prerequisites (the camera shall point to the railway track).

Visual Odometry:

- Measurement results (< maximum values) compared to reference
    - absolute distance: accuracy > 99.4% compared to GNSS / IMU data and to GTG
    - absolute speed: precision < 1 km/h compared to GNSS / IMU data
    - absolute distance between consecutive balises pairs: < 0.7% compared to the nominal distance stored in database

  within an estimated systematic uncertainty of 0.8% with the current setup. In other words, the results are in accordance with the reference.
- 3D train position is accurate on short scale (~few kilometres), but the drift accumulates and becomes relevant at higher distances.

Video Localisation:

- The accumulated drift in the calculated 3D train position can be reduced by referring to point-frogs.
- The precision, compared to GNSS / IMU considered as ground truth, is about 20 cm. In other words the precision of the localisation is within the accuracy of the ground truth.

Object detection:

- All AprilTags, located alongside the track, were successfully detected.
- Railway point-frogs were successfully detected in all points.

Current limitations:

- With the current setup, there is an issue in long tunnels with poor illumination, which can be solved by a slightly changed setup of the infrared illuminator.

- The performance in challenging weather conditions like heavy rain or fog have not been tested yet.

Options and future Improvements:

- Fixed installation position of the camera system in order to minimize systematic uncertainty of the train localisation
- Introduction of SLAM (**S**imultaneous **L**ocalisation **A**nd **M**apping) algorithms for drift compensation by creating a local map with the position of natural objects like bridges, trees or buildings and using them as landmarks.

Certifiability:

- Comparison of the results achieved by Video with already certified sensor axle counter
- Realtime capability and deterministic algorithms

For more details see chapter 6.3.1 to 6.3.3.

Sensor Technology FOS

Fiber optic sensing offers an interesting potential as a supporting technology for absolute localisation and for determining train length and speed. Again, even though the results were very good, there is still a lot of room for improvement.

Extent of use in railways:

- Fiber Optic Sensing offers great potential as a technology not only for train localisation but also for train speed, length, and integrity.
- Absolute measurements (no error accumulation)
- Continuous measurement of train movement
- Instantaneous snapshot of all trains moving on the track
- Determination of train front and rear ends and train speed
- Calculation of train length for determination of train integrity
- Very sensitive to vibrations and trains are easy to spot due to the high amplitude vibrations produced

Train localisation and train speed:

- Localisation error: 99% < 20 meter (gaussian distribution with standard deviation ~7.7m) using current parameters and comparing against GNSS
- Speed accuracy: 99 % < 2 m/s (gaussian distribution with standard deviation ~0.8m/s) using current parameters and comparing against GNSS

Train length and train integrity:

- Train length and integrity are accurately and continuously measured.
- Train length determination error: 87% < 20m

Current limitations:

- Localisation at train speeds below 25 km/h is still unsatisfactory.
- Determination of which track the train is coming from is still unsatisfactory.
- Bridges appear as a long channel (vibrate as a whole).
- Dependent on how the fiber optic cable has been laid out in relation to the tracks

- Dependent on what kind of material the fiber optic cable has been buried in

Options and future improvements:

- A hybrid thresholding model using both the power and entropy spectral flatness (ESF) should be used in order to eliminate interference from other objects completely.
- Possibility of using concurrent different models in real time in order to increase the confidence in the results
- Adjustments of parameters trading off accuracy by delay and/or noise and possibility of simultaneously running the analysis using these parameters in parallel and combining the results
- Better results can be achieved by using more detailed modelling with parameters determined by more reference runs with better clock synchronisation
- Real time analysis with reasonable processing power

Certifiability:

- Comparison of the results achieved by FOS with already certified sensor axle counter
- Realtime capability and deterministic algorithms

For more details see chapter 6.3.4.

SBB Innovation project Optical train localisation

The PoC showed that in principle deep learning-based algorithms can be used for optical train localisation. However, the algorithms still have to be refined and their reliability has to be further evaluated.

- Optical train localisation at different lighting and weather conditions is possible without the use of external infrastructure (if the DfA is loaded locally onto the train).
- Determination of track selective lateral position with a very high accuracy
    - > 90 % detection precision for most lightning conditions
    - < 70 % detection precision during night or at low visibility
- The described approach relies on the visibility of all adjacent tracks and kilometre panels
- Tracks around station entrances are critical and can impair the detection
- Poor recognition rate for mast boards due to alignment of the boards on the test route
- Good recognition rate (90%) for the kilometre panels
- Optical kilometre panel detection combined with optical track selection and DfA integration can be used without GNSS for full train localisation
- Concerning a possible certification, the algorithms have to be further tested and investigated.

In general: More data is needed to refine and further evaluate the optical train localisation approach.

2.4 Content overview

In the subsequent chapters the different localisation approaches are presented and evaluated as follows:

Chapter 3 describes and analyses approaches for relative and absolute train localisation by applying video technology and deterministic algorithms for visual odometry and object detection (e.g. AprilTags, points, etc.) based on a monocular camera setup.

In **chapter 4** the sensor technology FOS is described for the absolute localisation of moving trains in real time. The detection of their position, front and rear ends, corresponding length and velocity are all based on certifiable algorithms.

Chapter 5 considers a multi-sensor setup for accurate and safe localisation and the required functional architecture of such a sensor system enabling certification.

Chapter 6 documents the results of the measurement series and compares different technologies regarding accuracy, availability and ground truth, especially with respect to SIL4 axle counters amongst other approaches. However, the main challenge, that needs to be solved, is to ensure an exact and correct time synchronisation of all onboard and trackside components.

Chapter 7 discusses the SBB innovation project “Optical Train Localisation”. The study focusses on processing offline data from video streams and other sources using a deep-learning based algorithm approach for localisation.

3 Sensor Technology Video

The following sections describe the activities and results regarding the sensor technology video to meet the aims and objectives of section 2.1.

Section 3.1 gives a general overview and the definition of the terms used.

Section 3.2 describes the improvements and optimisation of the camera system.

Section 3.3 describes the procedure for the identification of the railway track and its application to the estimation of the camera extrinsic parameters, curvature of the track and detection of the point-frogs.

Section 3.4 describes the procedure for calculating the train position by Visual Odometry.

Section 3.5 describes the procedure for the detection of stopping plates and AprilTags in images collected by a dedicated camera with large focal length.

In section 3.6, the results of the analyses are shown. Different data runs are used to validate different use cases under different conditions.

3.1 Introduction

Images collected by a camera, that is located in the train and points to the railway track, can provide important information like train position and identification of track objects like railway points or stopping plates.

With the term *Visual Odometry*, the procedure for calculating the local position of the train, by comparing consecutive image frames, is meant. Such a method can be very precise on a short scale but it suffers of systematic uncertainties that accumulate over time causing a drift in the calculated position. With the term *Video Localisation*, the use of Visual Odometry, enhanced by the detection of objects with a fixed and exactly known position, is meant. Indeed, the precision of the calculated local position of the train can be improved by referring to objects like point-frogs, axle counters or AprilTags [3]. The goal is to reset the gradually increasing drift every time the above-mentioned objects are detected by the camera.

In the following, the procedure for calculating the train position is presented and it is based on deterministic algorithms only. The use of machine learning or artificial intelligent approaches are not suitable to reach a SIL4 certification as required for train localisation.

Data collected on 14th June 2019, along the track from Ostermundigen to Thun, were collected by two camera systems: one located in the locomotive and one located in the control wagon. Data from the locomotive have been extensively analysed and the calculated train position is compared to other sensor technology (See Section 6.3). In the following months, several additional measurements were recorded to validate different use cases under different conditions.

Table 3-1: Recorded video data

Abbr.	Direction	From	To	Date	Main focus
OT_1H	forward	Ostermundigen	Thun	14.06.2019	Localisation
OT_1R	backward	Thun	Ostermundigen	14.06.2019	Localisation
OT_2H	forward	Ostermundigen	Thun	14.06.2019	Localisation
OT_2R	backward	Thun	Ostermundigen	14.06.2019	Localisation
OT_3H	forward	Ostermundigen	Thun	14.06.2019	Localisation
OT_3R	backward	Thun	Ostermundigen	14.06.2019	Localisation
OT_4H	forward	Ostermundigen	Thun	14.06.2019	Localisation
OT_4R	backward	Thun	Ostermundigen	14.06.2019	Localisation
OT_5H	forward	Ostermundigen	Brig	14.06.2019	Localisation
Depot_1H	stillstand	Bern		03.12.2019	Camera calibration
BSG_1H	forward	Bern	St. Gallen	05.02.2020	Localisation with snow
BSG_1R	backward	St. Gallen	Bern	05.02.2020	Localisation with snow
BB_1H	forward	Bern	Brig	12.02.2020	Localisation with snow
BB_1R	backward	Brig	Bern	12.02.2020	Localisation with snow
BL_1H	forward	Biel	Lausanne	04.03.2020	Tilting train
BL_1R	backward	Lausanne	Biel	04.03.2020	Tilting train

The online data collection and offline processing are based on OpenCV [4], a C++ Library for Computer Vision.

3.2 Camera system

In this section, the system in use for image recording and storage is described. The system shall collect images in any weather and lighting conditions.

An infrared camera has been chosen to get brighter images with low lighting conditions, with respect to the images taken by a camera operating in the visible spectrum.

The camera exposure and sensor gain need to be controlled at run time and their values set accordingly to the lighting conditions.

In order to synchronize the image collected with the railway infrastructure, the timestamps from a GPS receiver is collected.

Timestamps and images are stored as raw data. In the last section, a real time compression algorithm is introduced to reduce the data storage without losing the information contained in the collected images.

3.2.1 Objective

The design of the camera system, presented in the following, is developed based on the experience maturated during the previous measurement runs (See Section 2.2).

Compared to the formerly used system, the main advantages and improvements are the following:

- A monocular camera replaced the stereo camera since the absolute scale can be determined also with a monocular camera by the so-called optical mouse tracking (See Section 3.4.2).
- An additional camera with a large focal length has been introduced to detect the AprilTags with high precision.
- A new set of camera parameters has been adjusted to deal with the dynamic lighting conditions of the train surroundings.

- A global time reference from GPS to synchronize the collected images with the railway infrastructure.
- A loss-less data compression to reduce the size of the data stored.
- Real time data processing for standstill detection of the train.

The mounting of the system allows an unexperienced user to setup the system for data collection.

The system supports a maximum of 6 hours of continuous operation and recording and is limited by the data storage and battery duration.

3.2.2 Camera box

The system for the image acquisition is composed as follows:

- Camera box containing
  - 2 IR Cameras (Model UI-3240CP-NIR-GL revision 2) with different focal lengths (8mm and 50mm) pointing towards the railway track (front camera) and pointing towards the side (tag camera) respectively. The front camera (8mm) is mainly used for relative localisation while the tag camera (50mm) is optimised to detect small objects like AprilTags that can be used for absolute position reference.
  - 2 NIR Illuminators (FLTT-808-1.8W-300m-CAP) for better illumination of dark scenes
- GPS Receiver (GlobalSat BU-353) to record the timestamp for the synchronization with the railway infrastructure.
- Rapid Prototyping Computer provided by Speedgoat. The processor is an Intel® Core™ i7-6700TE 8-cores, 2.40 GHz.
- Battery pack containing two batteries (Tattu 22.2V 15C 6S1P UAV Lipo) from 12000 and 16000 mAh each.

Figure 3-1 shows the system operating during the data recording in the area of Bern on 14th June 2019. As it can be seen in the picture, the connections between the camera box and the rapid prototyping computer were not fully integrated, making the installation in the locomotive not straightforward.

Figure 3-1 The camera setup operating on 14th June 2019 during the drive from Ostermundigen to Thun.

Figure 3-2 shows an improved camera box. Inside the box, the illuminators are connected to the TTL Converters which are powered from USB. The battery connector is connected to two DC/DC converters which convert the 24V input voltage into 12V for the illuminators. The cameras are powered directly over USB3.

The following interfaces of the box need to be connected:

- Battery connector to the 24 Volt battery
- USB3 (for Cameras) to the Rapid Prototyping computer
- USB (for Illuminator) to the Rapid Prototyping computer
- RS232 (for Illuminator) to the Rapid Prototyping computer

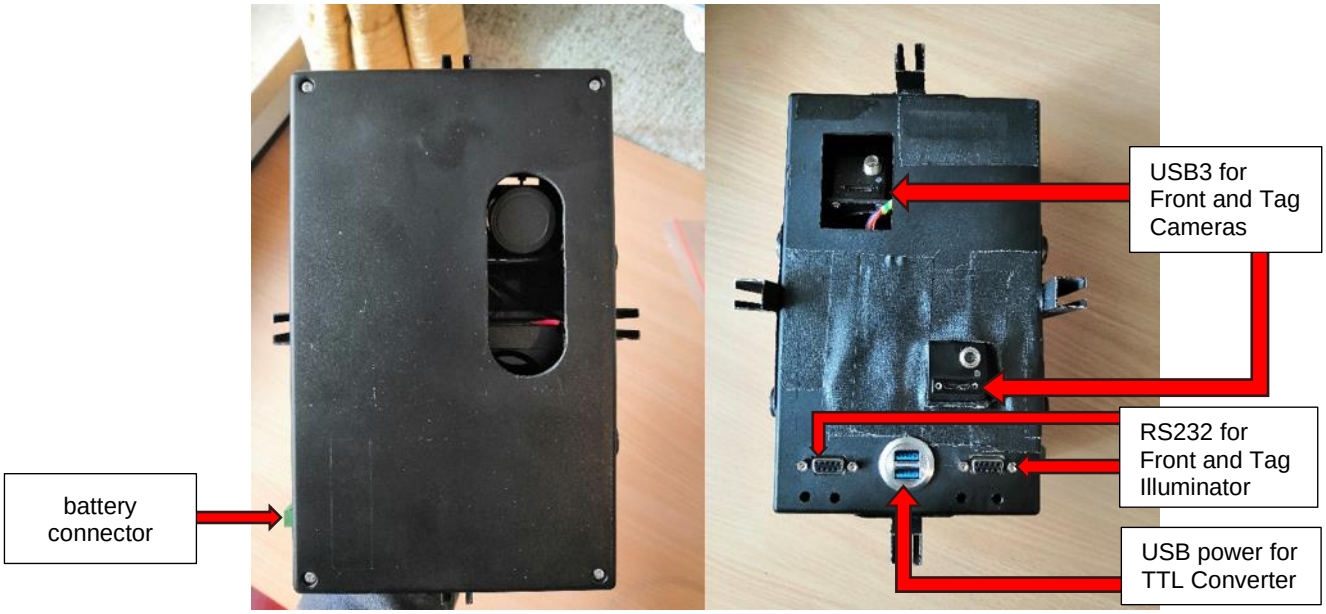

Figure 3-2 Front and Back view of the camera box

Different covers can be applied depending on the type of cameras and illuminators needed for the images to be taken. For example, if no AprilTags are located along the path, there is no need for the second camera and its illuminator.

Figure 3-3 shows the camera setup in operation on a RABDe 500, during the run from Biel to Lausanne (BL_1H and BL_1R).

Figure 3-3 The camera box mounted on a RABDe 500, during the run from Biel to Lausanne.

3.2.3 Camera settings

The IR Camera has different parameters that have to be adjusted for an optimal image collection based on the dynamic lighting conditions of the train surroundings. In the following, the relevant parameters are described.

- Shutter mode: global
  "On a global shutter sensor, all pixel rows are reset and then exposed simultaneously. At the end of the exposure, all rows are simultaneously moved to a darkened area of the sensor. The pixels are then read out row by row.
  Exposing all pixels simultaneously has the advantage that fast-moving objects can be captured without geometric distortions." [5]. This allows a simultaneous measurement of the position of the entire imaged area.

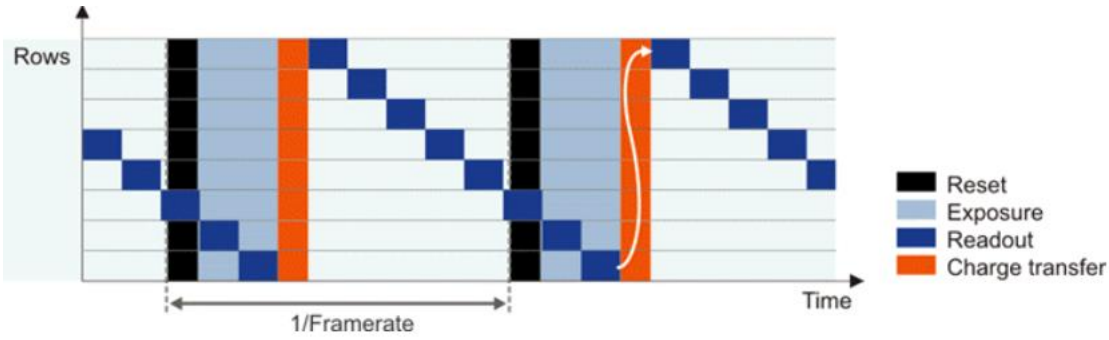

Figure 3-4 Global shutter sensor in live mode (Source: [5]).

- Pixel Clock: 86 MHz.
  It is frequency at which the sensor cells can be read out. Its value can be set between 7 and 86 MHz. The selected value is 86 MHz in order to reach the maximum frame rate.
- Frame Rate: 59.82 Hz.

The maximum frame rate is chosen to reach the highest possible resolution in the position estimation in the analysis.

- Image size: 1280x1024 pixels
  The maximum number of pixels is selected to acquire images with high resolution for the position estimation in post-processing analysis.
- Image bit depth: Grayscale with 10 bits.
  The increased data depth is necessary to cope with the high dynamic range in the scene illumination caused by possible shadow casts on sunny days or transitions in and out of a tunnel.
- Automatic functions:
  - Black Level Correction: auto.
    "The black level correction of the camera can improve the image quality under certain circumstances. By default, the sensor adjusts the black level value of each pixel automatically. If the environment is very bright, it can be necessary to adjust the black level manually." [5]. It is used for a maximum usage of the available gray-level depth in dependency on the scene illumination.
    For more info: https://en.ids-imaging.com/techtipps-detail/en_techtip-black-level.html
  - Automatic Exposure Shutter (AES): active (See Section 3.2.4)
    The control of the average brightness is achieved by adjusting the exposure. The brightness reference is controlled depending on the illumination of the railway track. Modification of the exposure does not amplify noise from the sensor in the acquired image.
  - Automatic Gain Control (AGC): active (See Section 3.2.4)
  - Brightness Reference: depending on the region of interest (See Section 3.2.4).
    The target brightness used as reference by AES and AGC.
  - Automatic Frame Rate (AFR): not used. Frame rate is fixed.
  - Maximum Exposure: 10 ms.
    Once the frame rate is fixed, the maximum exposure is determined (16.7 ms for the current settings). This value is further reduced to 10 ms in order to lower the motion blur in the image.
  - Control Speed: 100%.
    The speed of the control increments can be set in the range from 0 to 100%. The maximum value is selected in order to deal with rapid change of illumination due to tunnels or shadows.
  - Hysteresis: 10
    "The automatic control feature uses a hysteresis function for stabilization. Automatic control is stopped when the actual value lies in a range between (setpoint - hysteresis value) and (setpoint + hysteresis value). It is resumed when the actual value drops below (setpoint - hysteresis value) or exceeds (setpoint + hysteresis value). If the hysteresis value is increased, the control function will stop sooner. This can be useful in some situations." [5].
    The maximum hysteresis value is applied (default is 2) to collect a stable image brightness during transition between different sensor gain factors.
  - Log-mode: Auto (anti-blooming).
    "In Log-mode a threshold defines at which point the linear sensitivity pass over into a logarithmic characteristic. At very short exposure times (less than 0.1 ms) there may occur e.g. so-called crosstalk effects in the global shutter mode, which have the effect that the image content appears brighter in the vertical from top to bottom" [5].
    By using the option Auto, the automatic control of the anti-blooming is based on the set exposure time. It provides an additional extension of the perceived brightness range through logarithmic compression of very bright scene elements.

3.2.4 Image control in bright and dark scenes

The image acquisition needs to deal with the variable illumination of the scene the camera is pointing to. For example, shadow casts on sunny days or transitions in and out of a tunnel lead to a quick change in the lighting conditions the camera shall deal with. A fixed value of the exposure would result in images that are not well exposed in condition of high brightness dynamic range.

The camera exposure time and sensor gain are investigated and varied in order to get clear images for the calculation of the train position.

For practical purposes, the analysis presented has been tested on images containing AprilTags only. It shall be noted that the method proposed and the thresholds applied do not depend on the structure of the object on focus but on the illumination of the scene.

3.2.4.1 Impact of exposure time and sensor gain on the image brightness

Changing the exposure time increases the amount of light collected. This is the “natural” way to increase the brightness of the collected image.

Figure 3-5, Figure 3-6 and Figure 3-7 show the results of different exposure times in different conditions of illumination. As the illumination decreases, higher exposure times are preferred. Figure 3-7 a) shows that the amount of light collected is too high even with low exposure time. In this case, an even lower exposure time should be set. According to Figure 3-6, it is clear that with the maximum exposure (Figure 3-6 f), an image with the right brightness levels can be taken. In very dark scenes, as in Figure 3-5, increasing the shutter exposure time is not sufficient. In such a case, the sensor gain of the camera shall be increased.

It shall be noted that the maximum exposure time is limited by the desired frame rate that is set to a fixed value. Furthermore, an exposure time, that is too high, can cause motion blur in the collected image, when the projected point in the image illuminates multiple pixels in the image due to high velocity of the train. The motion blur is visible in Figure 3-8. The railway track in the image is not sharp due to motion blur likely caused by the high speed of the train combined with the high exposure due to the poor illumination of the scene.

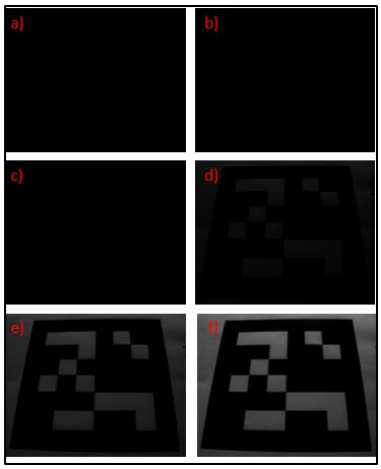

Figure 3-5 Images with very low illumination (50 lux) with different exposure times:
a) 0.5 ms b) 1 ms
c) 2 ms d) 5 ms
e) 10 ms f) 16.7 ms.

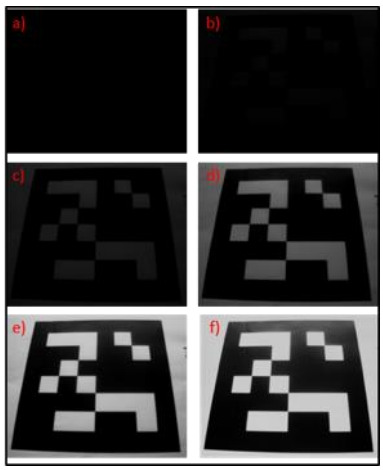

Figure 3-6 Images with low illumination (200 lux) with different exposure times:
a) 0.5 ms b) 1 ms
c) 2 ms d) 5 ms
e) 10 ms f) 16.7 ms.

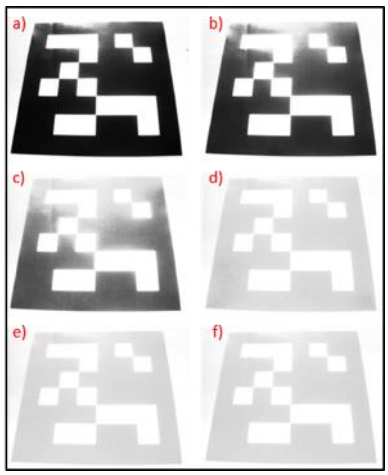

Figure 3-7 Images with high illumination (7000 lux) with different exposure times:
a) 0.5 ms b) 1 ms
c) 2 ms d) 5 ms
e) 10 ms f) 16.7 ms.

Figure 3-8 Image collected from a drive from St. Gallen to Bern on a cloudy day in winter. Due to the poor illumination of the scene, the exposure time was automatically set to a high value that causes motion blur resulting in an unsharp image.

Whenever the increase of the exposure time cannot lead to the desired image brightness, the sensor gain shall be increased, once the exposure time is set to the maximum value.

Figure 3-9 shows images taken at low illumination. Different sensor gain factors were manually applied in a) – e), while the camera automatic sensor gain control (AGC) is applied in f).

It shall be noted that the gain amplifies incoming signals from the scene as well as camera noise.

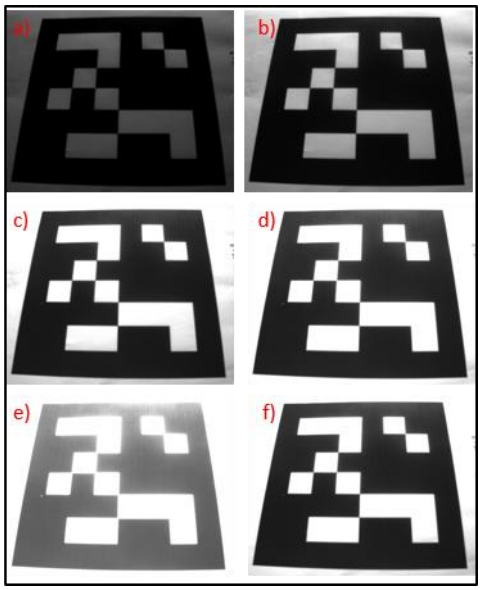

Figure 3-9 Images with very low illumination (50 lux) at fixed exposure time (16.7 ms). Different gain factors are applied: a) no gain, b) manually set to 33%, c) manually set to 66%, d) manually set to 99%, e) manually set to 99% and gain boost active, f) set to 84% by the automatic gain control.

3.2.4.2 Automatic control of exposure time and sensor gain

The automatic control for exposure time (AES) and sensor gain (AGC) of the camera adjust the exposure and sensor gain based on the brightness of the entire image. The target brightness is needed as reference for the control of automatic functions. This value can be increased (decreased) in case the scene is too dark (bright).

An automatic control function, that is based on the brightness information contained in the entire image, is not optimal and can cause errors in scenes with different illuminations (i.e. scene with shadows on the railway track). The scene under investigation can be restricted to a region of interest, i.e. the railway track regarding images collected by the front camera and the expected location of the object to be detected regarding images collected by the tag camera. For example, the AprilTags were placed on the catenary masts at a fixed height, which defines the expected location for object detection.

The solution adopted relies on the automatic control functions for exposure and sensor gain by using a brightness reference, which is not based on the entire image but on the above-mentioned region of interest.

Once the front camera is placed on the windscreen, a rectangular region of interest in the track area is determined automatically so that the track can be identified.

The region of interest of the tag camera is defined as a horizontal strip with constant height. The height is defined so that the AprilTag shall lie within the strip in the collected image.

Figure 3-10 (left) shows the image collected with AES active using the default value of the target reference brightness, that sets the exposure time to 7.2 ms. It can be seen that the region of interest (for practical purposes, a square instead of a strip containing the AprilTag is taken) is not sufficiently exposed.

Figure 3-10 (right) shows the image collected with AES active using an “adaptive” value of the target reference brightness based on the number of low-brightness pixels in the region of interest. The resulting exposure time was set to the maximum. The region of interest is well exposed.

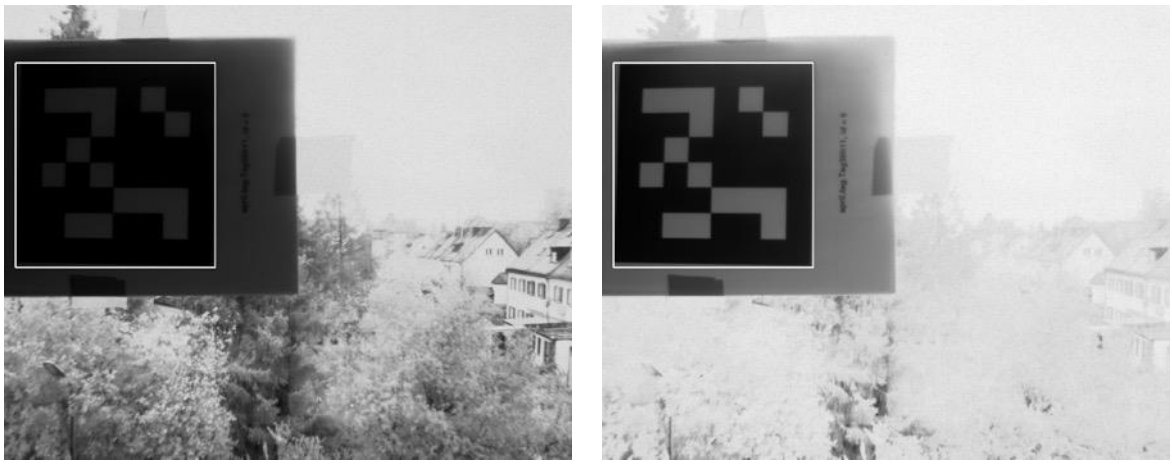

Figure 3-10 (Left) Image taken using the AES with default brightness reference value. The region of interest is too dark. (Right) Image taken using the AES with calculated brightness reference value. The exposure of the region of interest is sufficient.

3.2.5 Camera calibration

Ideally, the focal length of a camera is fixed by construction (8 mm for the front and 50 mm for the tag camera), but slight differences can be found from the nominal value. Similarly, a small deviation from

the nominal value of the principal point (located at the center of the image) occurs. The focal length and the principal point of the camera are measured by means of a calibration procedure.

The pinhole camera model is a mathematical relationship between a point in the real world and a point in the image plane. As is can be seen in Figure 3-11, in the image plane, the x values increase along the horizontal axis, the y values along the vertical one and the z values along the optical axis (i.e. the direction of the train motion).

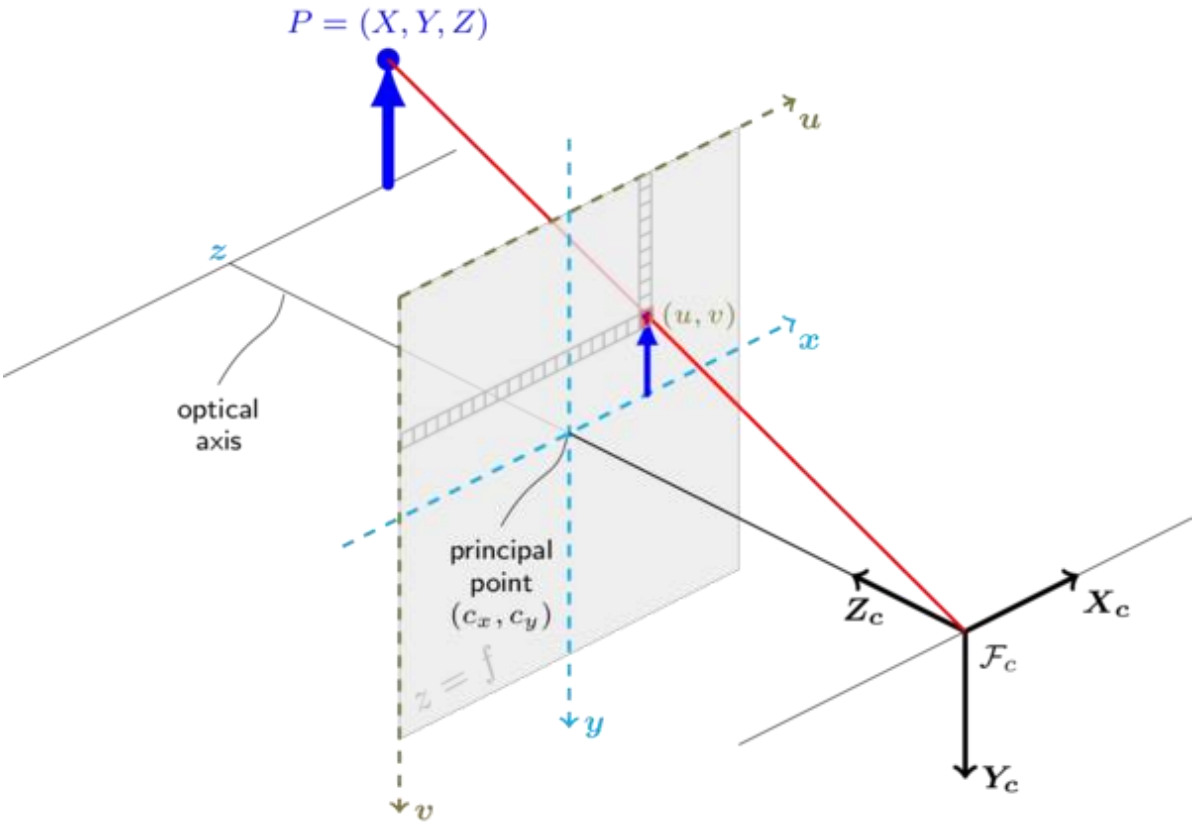

Figure 3-11 The camera pinhole model (Source: https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html).

The relation between the 3D coordinates of a point in the real world and a point in the collected image is given by the following formula:

$$s \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}}_{\text{intrinsic matrix}} \underbrace{\begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix}}_{\text{extrinsic matrix}} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

where *s* is the absolute scale, *(u,v,1)* are the homogeneous coordinates of the point in the image plane, *(X,Y,Z,1)* are the homogeneous coordinates of the 3D point in the real world, f_x and f_y are the coordinates of focal length, c_x and c_y are the coordinates of the principal point, *r* and *t* are the components of the rotation and translation of the camera reference system with respect to the real world.

The calibration of the camera allows for the measurement of the camera intrinsic parameters as well as the distortion coefficients [6] given a list of images collected.

Figure 3-12 shows a list of images collected for camera calibration. The calibration sheet (chessboard) was placed at different distances with different poses.

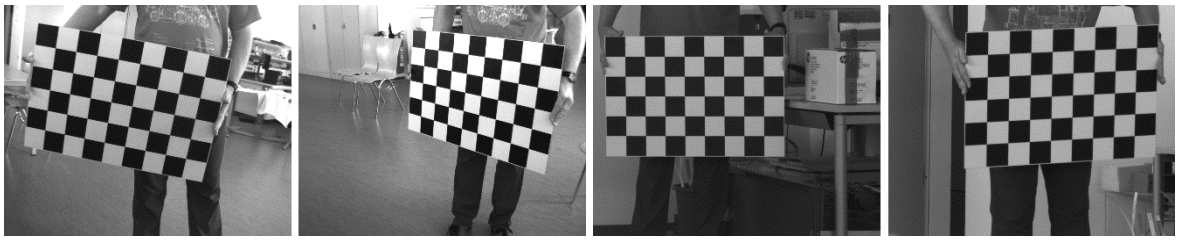

Figure 3-12 From left to right: Image for calibration of the two sets of front cameras and the two sets of tag cameras.

The Camera Calibration Toolbox for MATLAB® [7] is used as reference for the estimation of the intrinsic parameters and distortion coefficients.

Table 3-2 Parameters calculated by the calibration. The uncertainty in parenthesis refers to the last digits of the calculated value.

	Front (1)	Front (2)	Tag Old (1)	Tag Old (2)	Tag (1)	Tag (2)
f_x (px)	1523 (2)	1531 (2)	9402 (424)	9300 (267)	9487 (83)	9565 (82)
f_y (px)	1525 (2)	1533 (2)	9435 (437)	9346 (249)	9499 (81)	9562 (75)
c_x (px)	644 (2)	624 (3)	958 (332)	611 (0)	640 (0)	640 (0)
c_y (px)	512 (2)	485 (2)	606 (237)	-8 (0)	512 (0)	512 (0)
k1	-0.225 (3)	-0.218 (3)	0.57 (43)	0.08 (39)	0.24 (15)	0.31 (16)
k2	-0.143 (17)	0.119 (12)	-7 (25)	51 (28)	16 (41)	6 (43)
p1	-0.0007 (2)	-0.0003 (2)	0.03 (2)	-0.04 (1)	-0.001(2)	-0.004 (2)
p2	0.0006 (2)	0.0004 (3)	0.03 (3)	-0.009 (4)	0.0013(1)	0.002 (1)

The lenses used in the tag cameras have long focal length (50 mm) meaning that images are taken far away from the calibration sheet. This reduces the perspective and makes the light rays close to parallel, causing instability of the mathematical solution, especially when solving the lens distortion parameters k1, k2, p1, p2 [6], as can be seen in Table 3-2, where the error in the measurement of those parameters is extremely high. For this reason, the principal point c_x and c_y (marked in red) of the tag cameras was not included in the calibration procedure and the values were set to the optimal values, which is the center of the image with resolution 1280x1024 pixels.

3.2.6 Time synchronization for real-time analysis

In order to synchronize the data collected from the camera with the surrounding infrastructure, a time synchronization based on the GNSS global timing is needed.

For considerations about the latency of the system in use, refer to section 6.2.

3.2.6.1 Local time reference from camera system

Modern Linux operating systems provide several clock sources:

- Real time: clock, which should reflect the actual real time and is affected by discontinuous jumps in system time and by the incremental adjustments.
- Monotonic: clock, which never decreases but is affected by the incremental adjustments.
- Monotonic raw: hardware counter, usually the Time Stamp Counter (TSC) inside the processor, that is not affected by the incremental adjustments.

3.2.6.2 Global time reference from GPS

Satellites carry very stable atomic clocks, that are synchronized with one another and with the ground clocks.

A GPS Receiver is connected to the Rapid Prototyping computer through the USB port. The GPS receiver transfers the information in form of NMEA sentences [8] with a rate of 9600 bits per seconds.

The NMEA sentences contain geolocation and time information. The receiver can be configured in order to select only the NMEA sentences relevant for the time analysis. By selecting only part of the whole NMEA sentences, the latency in the GPS reception can be reduced. NMEA sentences of the ZDA, GGA and RMC types contain date and time information. Using the information carried by the GPS signal, a time resolution of one second can be reached.

The arrival time of the NMEA sentence depends on the length of the sentence, that is variable. The monotonic raw of the arrival time of the first byte of the NMEA sentence is saved together with the GPS time, since it does not depend on the above-mentioned factors and has a nanosecond resolution.

In case of poor GPS coverage (when the GPS receiver's antenna receives less than 4 satellites), a GPS fix is not received and the GPS time cannot be trusted. The presence of a GPS fix is saved to assess the quality of the received time during post-processing.

GPS data are analyzed in real-time in a dedicated Thread. The GPS time information is saved together with each frame, even if there is no fix.

3.2.6.3 Combination of global and system time

The following timestamps are saved together with the acquisition of the camera frame:

- Internal Image Time Counter (IITC): timestamp indicating the time of the complete image capture with a resolution of 0.1 microsecond.
- System Real Time (SRT): resolution of 1 nanosecond.
- System Monotonic Raw (SMR): resolution of 1 nanosecond.
- GPS Time (GPS): resolution 1 second.
- GPS Monotonic Raw (GMR): arrival time of the first NMEA sentence, resolution 1 nanosecond.

IITC, SRT and SMR have high resolution but they suffer of a small drift and need to be referred to a global time source. A combination of the system time information of the incoming image frame with the GPS global time is needed.

During initialization of the data acquisition code, the quality of the GPS signal (fix) is investigated. If the GPS signal is valid, the UTC time from GPS is taken as reference ($T_{0,GPS}$) and its arrival time as offset ($offset_{GMR}$). If the GPS signal is not valid, the system real time ($T_{0,SRM}$) is taken as reference and the system monotonic raw time as offset ($offset_{SMR}$).

For any incoming camera frame, the combined time is calculated and saved together with the image raw data:

$$t_{combi} = \begin{cases} T_{0,GPS} + t_{SMR} - \text{offset}_{GMR}, & \text{if valid GPS-Fix} \\ T_{0,SRT} + t_{SMR} - \text{offset}_{SMR}, & \text{if not valid GPS-Fix} \end{cases}$$

A possible case is, that during initialization the fix was not available and the GPS time is not used as reference for the calculation of the combined time. A fix could come after a while, so during post-processing a GPS-based combined time can be always be calculated.

Figure 3-13 shows the comparison between the GPS time (resolution of 1 second) with the calculated combined time (resolution of 1 nanosecond).

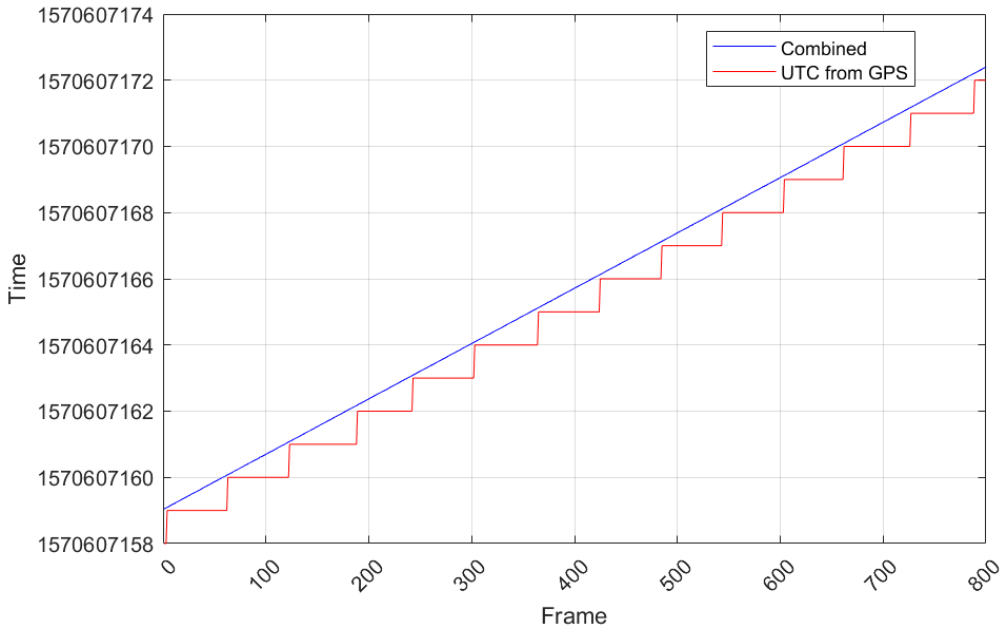

Figure 3-13 The UTC time from GPS (resolution of 1 second) is compared with the combined time (resolution 1 nanosecond).

3.2.7 Lossless data compression

The resultant 10-bit images occupy 2 bytes (16 bits) per pixel in memory. For storage, however, a simple pixel packing scheme where 4 pixels (40 bits) are packed in 5 bytes can be used in order to avoid this redundancy.

The advantage of using such a simple packing scheme (5 bytes for every 4 pixels) is its low resource utilization (high speed) and also the fact that every image uses exactly the same amount of storage (as long as all the images have the same dimensions) making it very easy to do random searches on a stream of images.

But in order to actually reduce the entropy of the image data, a lossless image compression algorithm should be used. For most “natural” images, compression ratios between 50% and 80% are to be expected, depending on the amount of noise present on the captured images. Lossless algorithms compression ratios are highly dependent on the amount of noise contained in the lower bitplanes of images, which is general not compressible.

Also, a “fast enough” algorithm should be applied in order to compress the video stream coming from the camera(s). In our case, each camera produces 1280x1024 gray level 16-bit images (2,621,440 bytes per image at 2 bytes per pixel) at 60 Hz. The system must be capable of compressing 60 fps per camera. It should also be pointed out that even compressed frames will be around a megabyte each (lossless compression) so the storage subsystem should also allow for the writing of at least around 60 megabytes per second per camera.

3.2.7.1 The FELICS algorithm

A fast implementation of the FELICS algorithm with tunable parameters for 16-bit pixels was developed and integrated into the solution.

The FELICS algorithm [9] is among the fastest lossless compression algorithms available and was used on the NASA Mars Exploration Rover embedded in the Solid-State Recorder (SSR) of the spacecraft.

It uses a single pass in raster order and uses the top and left pixel values in order to predict the current pixel and encode the difference between the predicted and actual values. Depending if the actual pixel is between the lower and higher values of its neighboring pixels, the error is encoded either using a range code (inside) or a golomb-rice code [10] with parameter *k*.

By choosing how to handle this parameter *k*, we can make the algorithm run faster than using a model to predict its value but compression may suffer. The user can choose how to handle this parameter in order to tune the speed and/or compression ratio of the algorithm.

Due to the varying size of the compressed images, it is not possible to have random access to a file composed of many concatenated images. In order to fix this, we have implemented a JFIF style byte stuffing [11] so that it is guaranteed that a certain byte value will never appear in the resulting byte stream except as a prefix for its immediately following byte. In this case, the byte composed of all set bits (255 in decimal or FF in hexadecimal) is replaced by the 2-byte sequence (FF 00 in hexadecimal).

Following the JFIF standard [12], the start of an image is signaled by the sequence FF E8 in hexadecimal and the end is signaled by FF E9, also in hexadecimal. As we substitute every occurring FF in the generated stream by a FF 00, it is not possible to encounter any other sequence that starts with FF in the stream unless we have actually requested it to be there.

This way, it is quite trivial to have random access to a file containing multiple images merely by searching for an FF E8 (start) and its following FF E9 (end).

Table 3-3 summarizes the results of the compression algorithm applied to drive OT_1H containing 57000 frames.

Table 3-3 The results of the loss-less compression.

Dataset	Raw image size (no pixel packing)	Raw image size (pixel packing)	Compressed image size	Compression speed
OT_1H	2621,440 KB	1638,400 KB	1057,584 KB	111 frames/sec

Even though the input image read from the camera is composed of 2 bytes per pixel (16 bits), the actual image is composed of only 10 bits per pixel, which makes its packed size equal to 1280 * 1024 * 10 / 8 = 1,638,400 bytes per frame. Using this as the original uncompressed size, the algorithm reduces the amount of storage by a factor 0.6 on average. For the test system, the compression speed was measured to be around 111 frames per second which allowed it to compress the images to be stored when using a single camera, which requires at least a compression performance of at least 60 fps. The performance of the algorithm has been tested on the Rapid Prototyping computer (See Section 3.2.2).

3.3 Railway identification and extrinsic camera calibration

The identification of the railway track in the collected image is described in section 3.3.1. Three algorithms are evaluated in order to identify the track in every collected image frame.

The position of the railway track in the image is the fundamental brick for several analyses:

- Estimation of the camera extrinsic parameter (Section 3.3.2): as described in section 3.2.5, the determination of the camera extrinsic parameters is one of the steps required to calculate the position of an object in the real world from the collected image. An automatic procedure for the estimation of the camera extrinsic parameters from the position of the track in the image, is described.
- Pitch detection and compensation (Section 3.3.3): the pitch of the train can be determined from the position of the track in the collected image. Variations of the pitch, corresponding to steep increases or decreases of the train speed, can be compensated.
- Detection of railway points (Section 3.3.4): the precision of the calculated local position of the train can be improved by referring to point-frogs that have a fixed and exactly known position. Railway points can be identified based on the intersection of the main track with the side tracks and the position of the point-frog in the image can be determined.
- Curvature of the track (Section 3.3.5): from the position of the railway track in the collected images, the yaw rate can be determined. By combining the yaw rate with the train speed, the curvature of the track is determined.

3.3.1 Automatic track detection

The railway track driven on by the train can be identified in the collected image, assuming that the camera points to the railway track. The procedure explained in the following is the fundamental brick for a correct estimation of the camera intrinsic parameter (See Section 3.3.2).

The first step is to find gradient edges in the collected image. Once the image is processed, the line detection algorithm is applied to match the edges in order to form a line.

Three different algorithms are investigated. For each algorithm, the parameters have been varied and the optimal values and thresholds are chosen in order to identify the railway track.

3.3.1.1 Canny contour detection and Hough transform

The OpenCV function `Canny` [13] is used to spot contours within the image. The following parameters are used:

- Lower (upper) threshold of the hysteresis procedure set to 100 (300).
- Aperture for the gradient operator set to 3.

The OpenCV function `HoughLinesP` [14] is then applied with the following parameters:

- Accumulator threshold parameter set to 300,
- Minimum line length set to 300,
- Maximum allowed gap between points on the same line to link them set to 200.

Figure 3-14 shows the images after the processing steps. The leftmost figure is the original picture, followed by the picture after applying the Canny edge detection (Figure 3-14 middle) and after that, the picture with the detected lines superimposed (Figure 3-14, right). The lines in yellow are the lines detected from the algorithm while the lines in green are the two identified as belonging to the railorad track.

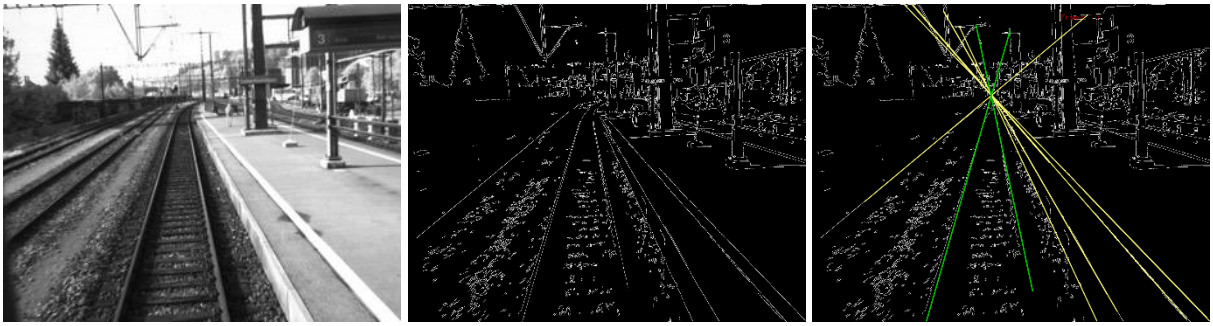

Figure 3-14 (left): Original picture, (middle): Picture after Canny contour detection, (right): Picture after HoughLinesP line detection

3.3.1.2 Morphological skeleton and Hough transform

The goal of the morphological skeleton is to reduce the redundant content of an image to known shapes with 1-pixel width.

The process involves two steps:

1. Image thresholding with adaptive threshold to reduce the grayscale image (Figure 3-15 left) to a binary one (Figure 3-15 right). The OpenCV function `adaptiveThreshold` [15] is not used, since it adapts the threshold based on the information of neighboring pixels. A fixed threshold based on the brightness of the image region in front of the train is applied.
2. Iterative erosion and dilation in order to have the skeleton image (Figure 3-16).

The OpenCV function `HoughLinesP` is then applied with the following parameters:

- Accumulator threshold parameter set to 200,
- Minimum line length set to 300,
- Maximum allowed gap between points on the same line to link them set to 500.

The identified track lines are shown in green in Figure 3-16 (right). The yellow line (difficult to be seen in the figure, since it is very close the green one) is a detected line not associated to the track (it is probably the outer part of the track).

Figure 3-15 (left): Original picture, (right): Picture after adaptive threshold

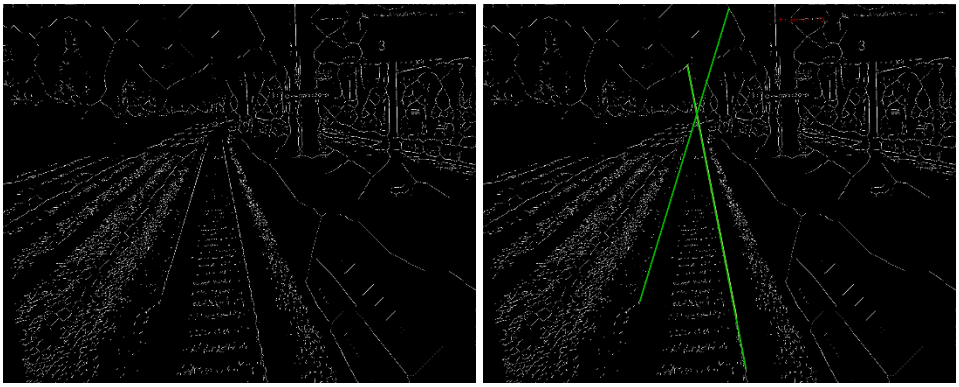

Figure 3-16 (left): Picture after iterative erosion and dilation, (right): Picture after applying the HoughLinesP line detection

3.3.1.3 Fast line detector

The OpenCV class FastLineDetector [16] based on [17] has been tested.

The class performs first an edge detection using Canny and then identifies line segments from the resulting image. The parameters applied are listed in the following:

- Lower (upper) threshold of the hysteresis procedure set to 100 (300).
- Aperture for the gradient operator set to 3.
- Minimum length of the calculated segment set to 100 pixels
- Minimum distance between the expected and the measured segment point to be considered an outlier set to 1.4 (default value).
- Incremental merging of line segments deactivated.

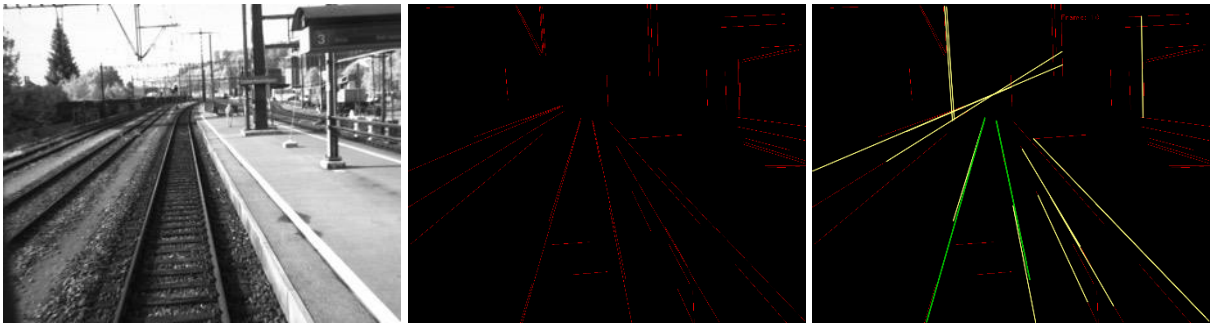

Figure 3-17 (left): Original picture, (middle): Segments found by the Fast line detector, (right): Lines found by the Fast line detector.

The fast line detector was found to give the highest track detection efficiency with lowest misidentification rate among the algorithms taken into consideration.

3.3.1.4 Procedure for track identification

The identification of the railway is divided into two parts: The first part is covered, when the train is at rest, the second part takes place, when the train is in motion.

The goal of the first part (initialization) is the determination of the starting point of the railway track situated at the bottom of the image and the estimation of the camera extrinsic parameters (Section 3.3.2) with the train at rest. This phase ends, once the tilt – pitch – yaw angles are estimated with a desired precision. In case the angles cannot be estimated with a desired precision, an acoustic signal will alert the user, so that the camera shall be placed elsewhere. The same happens in case of occlusion of the railway track in the image.

The goal of the second part is the identification of the driven railway track and side tracks when the train in motion (Section 3.3.4).

3.3.2 Automatic extrinsic camera calibration

The camera extrinsic matrix can be defined by three Euler angles (pitch, yaw and tilt according to Figure 3-18). The pitch is defined as the rotation angle around the side-to-side axis. The yaw is defined as the rotation angle around the vertical axes. The tilt is defined as the rotation angle around the front-to-back axis (the focal axis).

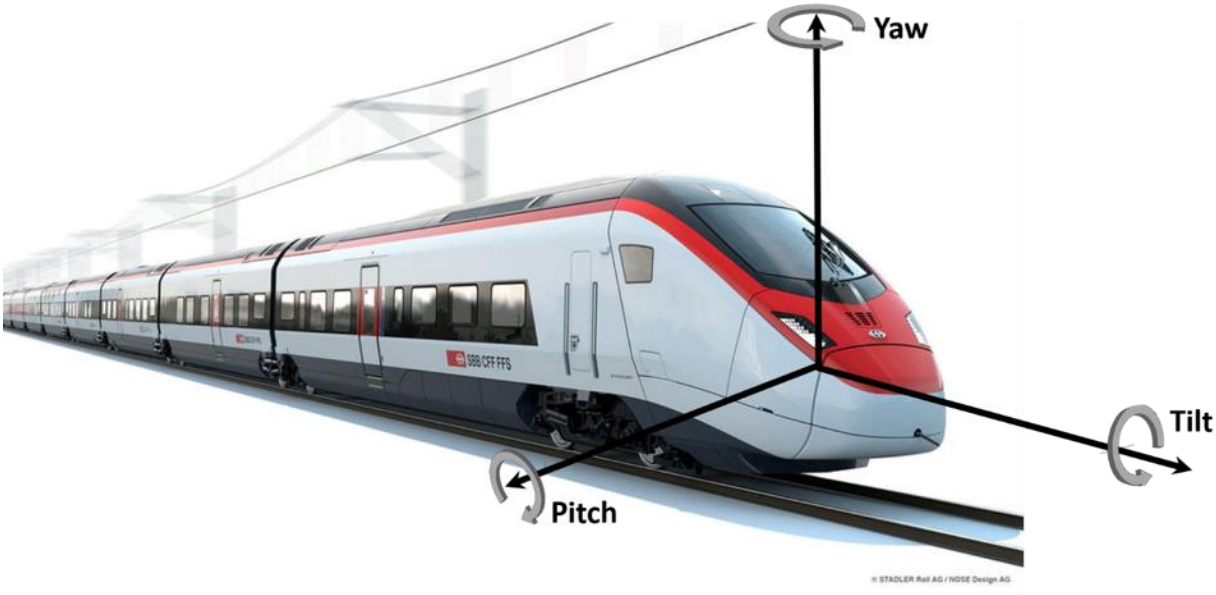

Figure 3-18: Sketch of the Stadler EC250 with the reference system that shows the yaw, pitch and tilt rotations around the axis (Source: https://bahnblogstelle.net/2017/03/28/stadler-ec250-schweizer-zughersteller-praesentiert-neuen-hoch-geschwindigkeitszug-video/).

In this section, the procedure for the estimation of pitch, tilt and yaw based on the image of the railway track is described.

3.3.2.1 Pitch and Yaw estimation

The method proposed is based on the calculation of the vanishing point (VP) of the railway track in the camera image. The vanishing point is defined as the point on the horizon line where parallel lines converge. This point depends on pitch and yaw angle. The point does not give any information regarding the tilt angle. This can be intuitively clarified, since rotating the camera around the focal axis does not change the vanishing point of the railway track.

The yaw angle α and the pitch angle β are estimated using the following formula:

$$\alpha = -\arcsin\left(\frac{V_x - c_x}{\sqrt{(V_x - c_x)^2 + (V_y - c_y)^2 + f^2}}\right), \quad \beta = -\arctan\left(\frac{f}{V_y - c_y}\right)$$

where V_x and V_y are the coordinates of the detected vanishing point in the image, c_x and c_y are the coordinates of the principal point and f the focal length.

Figure 3-19: shows the detected railway track (green) and the intersection, that defines the vanishing point (cyan).

Figure 3-19: The detected railway track (green) and the vanishing point (cyan) are shown in the image collected.

3.3.2.2 Tilt estimation

The tilt is estimated by selecting a region in front of the train centered on the railway track. The region is rectified by applying the calculated pitch and yaw and transformed to gain a bird-eye top-view.

Figure 3-20: shows the result of the track detection algorithm applied to the recognition of the sleepers (red segments). As it can be observed, it is difficult to identify the sleepers since the image gradient is poor. The identification strongly depends on the illumination of the scene and on the shape of the sleepers, that cannot be guaranteed to be smooth. In order to measure the tilt with high precision (less than 1 degree), a direct measurement from the sleepers is not suitable.

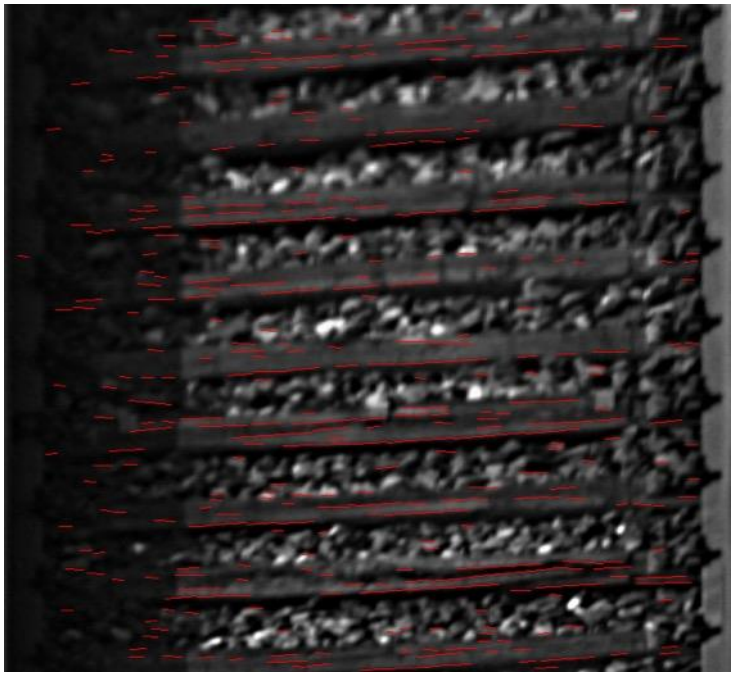

Figure 3-20: A bird-eye view of the railway track is shown. The sleepers (red segments) are difficult to be identified by the fast line detector algorithm.

The railway track width in the image slightly depends on the camera tilt. A subpixel precision would be required in order to measure small camera tilt directly from the track width. A possible solution could be to measure the track width in the image rotated by large tilt angles, where the track width changes significantly. A fit of the measured track widths with a cosine function should give an estimation of the camera tilt.

Figure 3-21 shows the images transformed according to a tilt angle of -45° (left), 0° (center) and 45° (right).

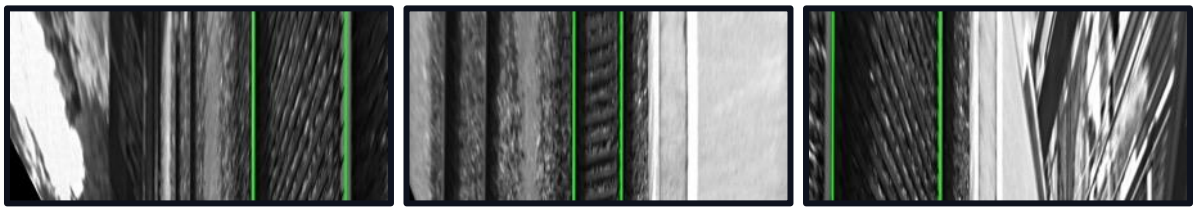

Figure 3-21 The bird-eye view image is rotated by a 45° (left) , 0° (center) , 45° (right) tilt angle and the track is identified.

The fit algorithm gives a tilt angle of 4°, that is not in agreement with a rough empirical estimation.

In order to analyze the collected data, an empirical evaluation of the tilt angle can be performed by comparing the original image with the rectified one. By constraining the rectified image to have sleepers parallel to the bottom of the image, a rough estimation of the tilt can be given.

Figure 3-22 (left) shows the image from a bird eye view rectified only for yaw and pitch. Figure 3-22 (right) shows the image rectified with the addition of a tilt angle of ~ 2.9 ± 0.2 degrees estimated empirically.

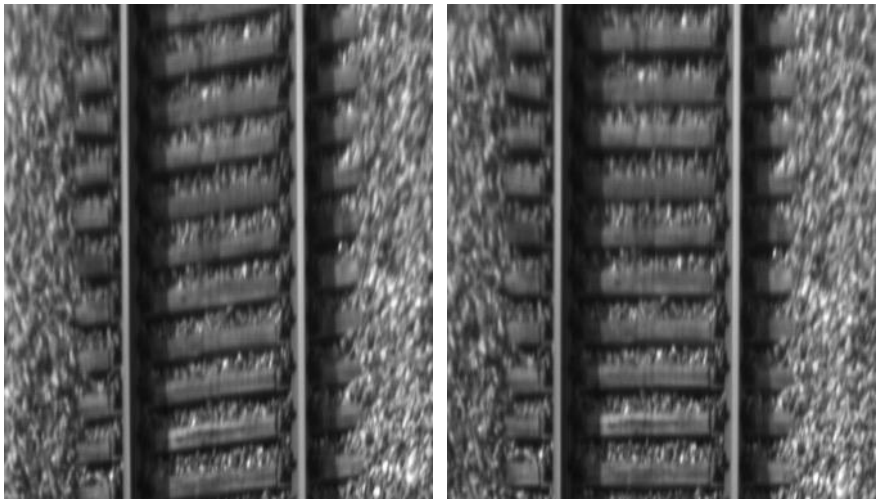

Figure 3-22 (left): The image rectified for pitch and yaw estimation and transformed to a bird-eye view. The sleepers are not parallel to the bottom of the image, meaning that the tilt shall be corrected, (right): By correcting the tilt of the camera, the sleepers are parallel to the bottom of the image.

3.3.3 Pitch detection and compensation

As described in section 3.3.2.1, the pitch can be estimated, once the vanishing point is found in the image.

Figure 3-23 shows the difference (in blue) of the pitch, measured in all the image frames, with respect to one measured during the calibration, where the train was at rest. In red, the train speed is displayed. Negative (positive) values of the pitch difference correspond to a camera looking up (down) with respect to the initial pose. Currently, with the actual precision in the measurement of the track position, it is not possibile to correlate the change of the pitch to a change of the train speed. However, it can be observed that, as soon as the drive begins, the pitch difference tends to negative values (the red dotted line highlights the frame when the train starts).

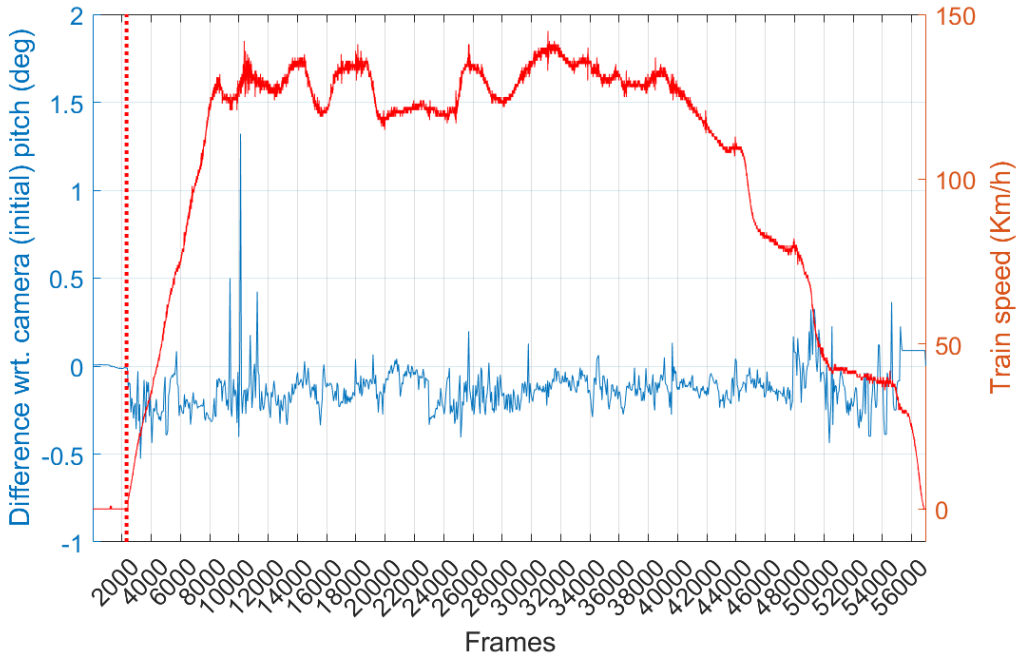

Figure 3-23: The difference (in blue) of the pitch measured during the whole measurement run, with respect to one measured during the calibration in the initial frames, where the train was at rest. In red, the train speed is displayed. The red dotted line highlights the frame when the train started.

With a more precise track identification, the calculated pitch difference can be used to correct the calculated train position.

3.3.4 Railway point detection

The identification of railway points in the image can be used as global reference for the calculation of the train position with visual odometry (results see section 6.3.3).

For the track identification during drive, the images collected from the camera are rectified by applying the camera extrinsic parameters measured in the initialization phase and transformed to obtain a bird-eye view in a region of interest in front of the train.

Possible railway track candidates are detected using the line detection algorithm described in Section 3.3.1 with the constraint that the track width shall be close to the track width measured in Section 3.3.2 (initialization phase).

Figure 3-24 (left) shows the collected image with the selection of a region of interest (in yellow) in front of the train 6 meters long and 6 meters wide. The bird-eye view of the selected region is shown in Figure 3-24 (right) where the detected lines are highlighted: the driven track is found and shown in green, while a side track is detected and shown in red.

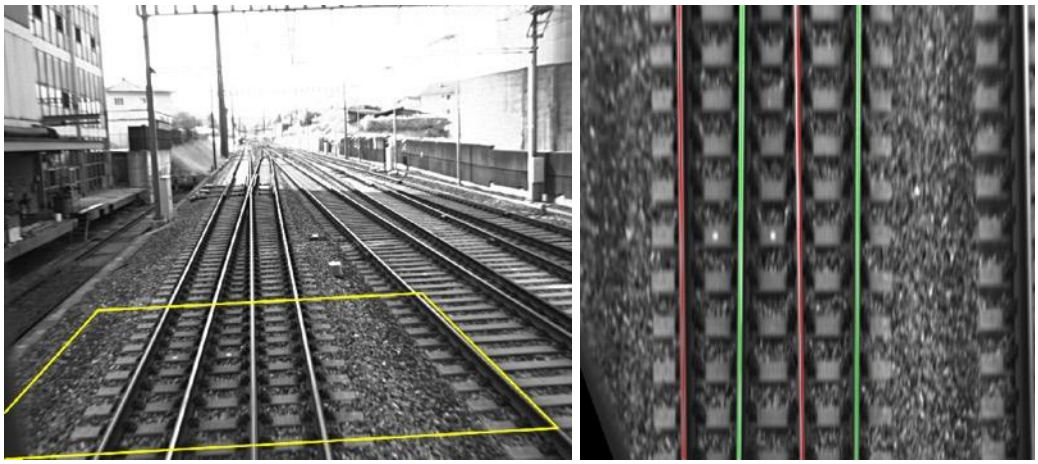

Figure 3-24 (left): A region of interest of 6x6 meters is defined in the collected raw image, (right): the region of interested is rectified and transformed according to a bird-eye view and the main track (green) and a side track (red) are identified.

The performance of the identification of the driven track and the point recognition were tested using the dataset OT_1H. The driven track was idenfitied in 99.4% of the frames.

3.3.4.1 Point-Frog detection

The intersection between the main with the side track defines the frog of the point. Figure 3-25 (left) shows the detection of the frog (circle in cyan), where the driven track (green) and side track (magenta) intersect.

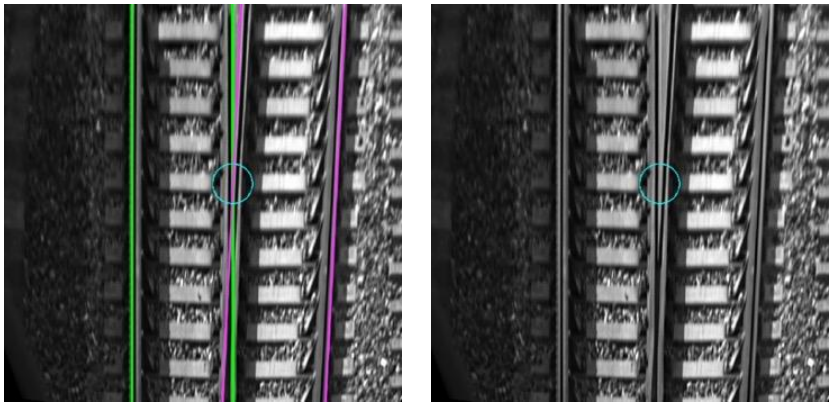

Figure 3-25 Left: The point-frog (circle) is identified from the intersection of the main track (green) and the side track (magenta). Right: A small deviation (~1 meter) between the identified and the truth point-frog is observed.

The performance of the point recognition is evaluated in the following. Table 3-4 shows the type of points, that are visible in data collected, and the result of the detection. The performance of the detection of curved points could not be tested since they were not present in the driven path.

Table 3-4: Results of the point detection. The latitude, longitude and ID of the points are taken from the topology database (DfA). The column *Frame ID* refers to the collected image frame, where the train passes the detected point. The column *N Frames* refers to the number of frames where the frog is detected. In most of the cases, the number of frames a point frog is detected increases the precision of the measurement of its position. * the point was detected but its position deviates (> 5 meters) from the truth.

Point ID	Type	Latitude	Longitude	Detected?	Frame ID	N Frames
OST 19	Einfache Weiche	46,9533	7,4826	yes	4813	26
OST 20	Einfache Weiche	46,9529	7,4827	yes	5060	23
OST 55	Einfache Weiche	46,9483	7,4861	yes	6593	16
GUE 1	Einfache Weiche	46,9381	7,4985	yes	9487	11
GUE 4	Einfache Weiche	46,9364	7,5017	yes	9898	9
GUE 9	Einfache Weiche	46,9359	7,5026	yes	10134	10
GUE 31	Einfache Weiche	46,9335	7,5068	yes	10813	10
GUE 28	Einfache Weiche	46,9324	7,5088	yes	11129	10
GUE 47	Schnellfahrweiche	46,9317	7,5101	yes*	11401	5
GUE 48	Schnellfahrweiche	46,9307	7,5118	yes*	11502	11
GUE 52	Einfache Weiche	46,9288	7,5151	yes	12058	11
GUE 55	Einfache Weiche	46,9276	7,5173	yes	12548	10
RUB 1	Einfache Weiche	46,9037	7,5429	yes	18074	10
RUB 12	Einfache Weiche	46,8947	7,5481	yes	19773	11
MS 1	Einfache Weiche	46,8803	7,5573	yes	23005	11
MS 18	Einfache Weiche	46,8710	7,5600	yes	24818	12
MS 27	Einfache Weiche	46,8695	7,5605	yes	25093	11
WCH 1	Einfache Weiche	46,8468	7,5672	yes	29445	10
WCH 5	Einfache Weiche	46,8444	7,5679	yes	29873	10
WCH 14	Einfache Weiche	46,8421	7,5686	yes	30242	9
WCH 20	Einfache Weiche	46,8375	7,5699	yes	31017	10
UTI 1	Einfache Weiche	46,8003	7,5810	yes	37900	10
UTI 8	Einfache Weiche	46,7910	7,5837	yes	39527	10
UTI 11	Einfache Weiche	46,7906	7,5839	yes	39679	10
TH 119	Kreuzung	46,7604	7,6208	yes	49853	28
TH 131	Kreuzung	46,7595	7,6225	yes	50788	31
TH 181	Einfache Weiche	46,7567	7,6267	yes	52936	33
TH 183	Kreuzung	46,7565	7,6270	yes	53153	27
TH 185	Kreuzung	46,7557	7,6279	yes	53752	35
TH 187	Kreuzung	46,7554	7,6282	yes	53979	34
TH 198	Kreuzung	46,7551	7,6286	yes	54210	37
TH 219	Kreuzung	46,7548	7,6289	yes	54473	35
TH 221	Kreuzung	46,7545	7,6292	yes	54826	56
TH 224	Kreuzung	46,7543	7,6295	yes	54901	39
TH 226	Einfache Weiche	46,7540	7,6298	yes	55081	36

Currently, the position of the frog in the high-speed points from frames 11388 and 11470 is difficult to detect. Indeed, the intersection of nearly parallel lines is difficult to be detected. As it can be observed in Figure 3-26 (right), the algorithm detects a frog that does not correspond to a frog in the track (Figure 3-26 (left)).

Figure 3-26 (left): The image of a high-speed point is collected, (right): A bird-eye view of the high-speed point, where both main (red) and side (green) tracks are detected. The tracks are almost parallel and the intersection is difficult to measure.

Several factors affect the measurement of the position of the point-frog and are listed in the following:

- Type of points: As shown, the uncertainty of the point-frog position increases with the point crossing angle. Point-frogs of nearly parallel lines are detected with low precision.
- Track identification algorithm: The detection of the point-frogs depends on the identification of the track. The algorithm can be refined and the precision in the identification need to be estimated.
- Camera calibration: The precision in the measurement of the camera extrinsic parameter affect the measurement of the position of point-frog.
- Absolute scale: The conversion factor from image pixel to distance in the real world has an impact in the measurement of the point-frog position.

The camera extrinsic parameters and the absolute scale can be measured with high precision.

Currently, only an approximative estimation can be given. As it can be seen in Figure 3-25 (right), the precision in the detection of the point-frog is about 1 meter in the (best) case of a simple switch-point.

A detailed estimation of the precision of the point-frog position need to be performed in the next step.

3.3.4.2 Detection of the tongue position

The position of the tongue rail determines the direction of the train.

Figure 3-27 shows a tongue rail in the image collected. The lines in the crossing can be identified (magenta), but at the moment it is not possible to clearly determine the state (open or close) of the incoming tongue.

Figure 3-27: (left): The image of a crossing point is collected, (right): A bird-eye view of the crossing point with detected lines (magenta). The tongue rail can be hardly identified among the detected lines.

3.3.5 Detection of the curvature of the track

As described in Section 3.3.2.1, the yaw can be estimated once the vanishing point is found in the image. The curvature (inverse of the radius of curvature) is given by the ratio of the yaw rate and the tangential velocity (the absolute speed in Figure 6-5 (left)). Figure 3-28 shows the curvature calculated from the track position in the image (blue). The signal shape of the measured curvature seems to be in accordance with the signal shape of the curvature stored in the database (red). However, a difference of the curvature values is observed at every local peak. For a detailed comparison, a more accurate identification of the railway track position with the camera system is required.

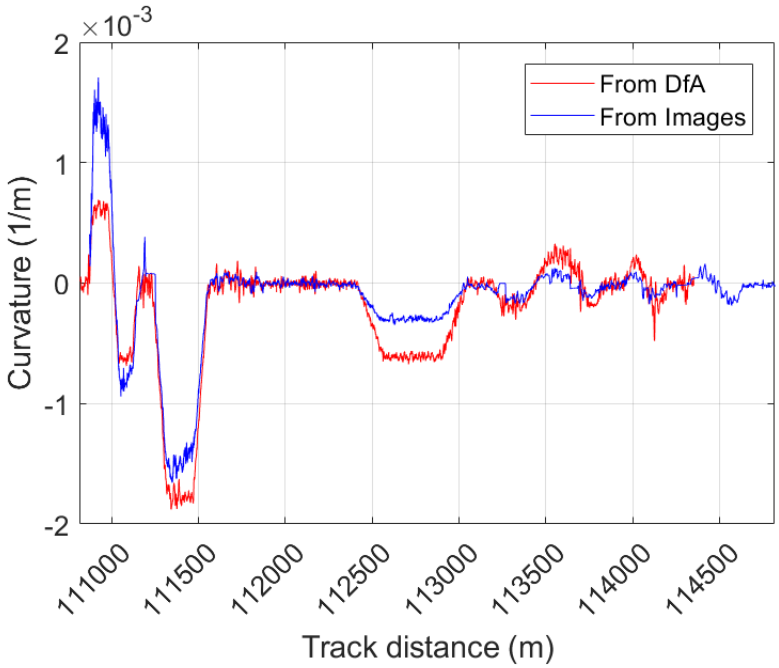

Figure 3-28 The estimated curvature, derived from the position of the railroad track in the image, is compared to the curvature stored in the database for a small section.

3.4 Train localisation by Visual Odometry

In this section, the procedure for calculating the train position from the collected images is described.

The absolute distance is measured by means of the so-called optical mouse tracking, where brightness levels between consecutive frames are compared and the pixel shift is estimated. The absolute scale, i.e. the conversion factor between distances in the image plane (measured in pixels) and distances in the real world (measured in meters), is calculated by taking the track width as a reference.

The 3D position of the train is estimated by using the optical flow between consecutive frames. Features (points in the image with high brightness gradient) are extracted from the image and tracked from one frame to the next one. This technique provides the rotation matrix and translation vector (pose) of the train motion with respect to the railway track. The translation vector is normalized since the absolute distance cannot be determined at this stage.

By combining the results of the measured pose with the measured absolute distance, the 3D position of the train can be calculated.

The calculated train position can be improved by freezing the reference frame (key-frame) for a number of following frames. This procedure improves the accuracy of the estimation of the direction of motion, in particular when the train speed is low.

3.4.1 Objectives

The goal of this section is the calculation of the relative train position with Visual Odometry. The relative train position refers to the position of the train with respect to the railway track. High precision is expected to be achieved with the sample rate (60 Hz) and image resolution (1280x1024 pixels) of the camera in use. The calculated distance is expected to have high resolution on a short scale (~1 km). It shall be noted that, since a relative, instead of global, position is calculated, the systematic uncertainties accumulate over time causing a drift of the calculated position, that becomes relevant at high distances. The drift can be compensated by referring to fixed objects like point-frogs, axle counters or AprilTags with a fixed and exactly known position (see section 6.3.2)

3.4.2 Determination of the absolute distance with the optical mouse tracking

The use of a monocular camera for visual odometry has the limitation that distances cannot be measured using image information only. In order to measure distances from images, the absolute scale, i.e. conversion factor between the image pixels and meters in a rectified planar image, is determined with the so-called mouse tracking technique, in analogy with the optical computer mouse, that detects movement relative to a surface by detecting the emitted light.

3.4.2.1 Template matching for optical mouse tracking

The absolute distance in pixels is determined using the template matching technique implemented in OpenCV [18].

The following areas in the image are defined for the template matching:

- Search area (Figure 3-29a) in blue: this is the window, where the template matching is performed
- Template area (Figure 3-29b): this is the template region used to find matches within the search area.
- Search area rectified (Figure 3-29c).

The search and template areas are rectified using the parameters estimated in Section 3.3.2 and an homography transformation has been applied to get a bird-eye view (Figure 3-29b and Figure 3-29c).

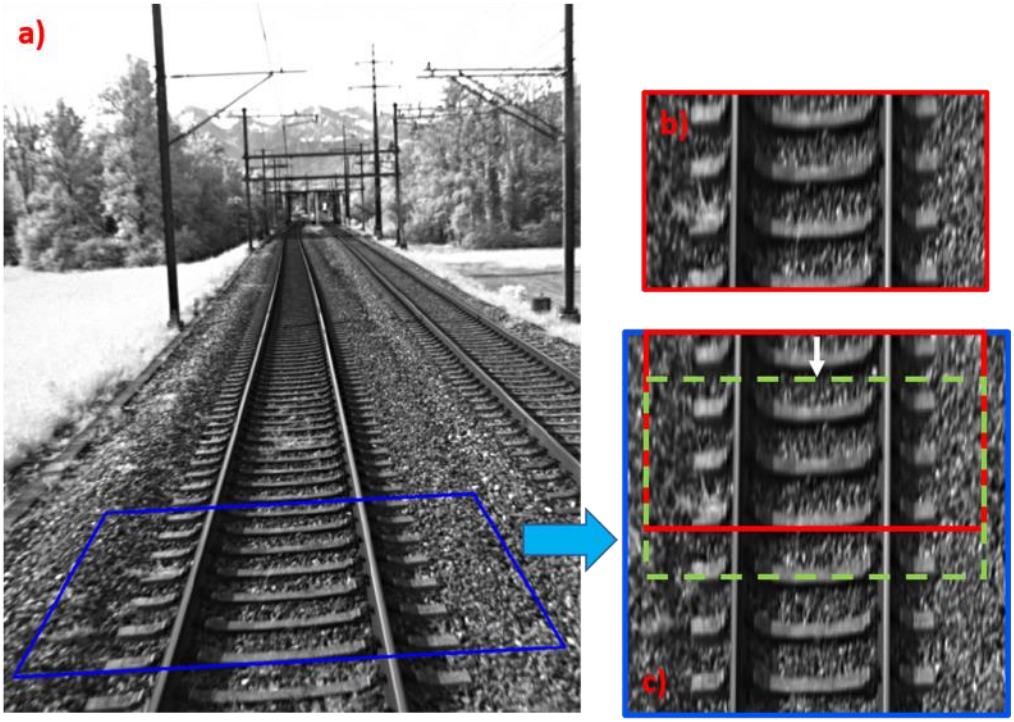

Figure 3-29 a) A search window (blue) is selected in the collected image, b) a template image (from the previous frame), c) the search window is rectified and transformed according to a bird-eye view. The image template b) slides through the search window and the best match is found (green) The pixel displacement (white arrow) between the top of the search image and the position of the green template is proportional to the travelled distance.

In Figure 3-29c, the template image of the previous frame (red) slides through the actual search window (blue) and the position of the best match (green) based on the intensity values is found by minimizing the normalized sum of square difference. The difference in pixels between the red and green images is proportional to the traveled distance.

3.4.2.2 Track width as reference for absolute scale determination

The absolute scale, i.e. the conversion factor between image pixels and distance in the real world, is obtained by taking the track width as a reference.

Figure 3-30 (left) shows the width of various railway tracks in the world. The width of 1435 mm is the one, that is by far the most in use and it is called standard gauge. All the railway tracks handled by SBB are built according to the standard gauge. The profile of the rail from SBB is shown in Figure 3-30 (right).

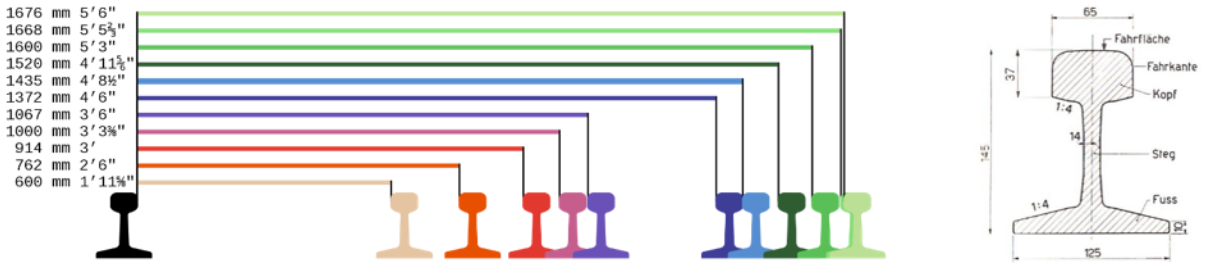

Figure 3-30 (left): The size of the railroad track width are listed. The standard gauge (1435 mm) is widely used in world. (right): the profile of the track.

The deviation of the railway track width and its uncertainty in knowledge has a direct impact on the calculated driven distance. Although the size of the standard gauge is fixed, some deviations to that

value are allowed. According to the railway regulation in Switzerland, the lower limit to the track width is 1430 mm while the upper limit is 1470 mm, including the gauge widening.

The track width is measured by SBB twice a year. Figure 3-31 shows the distribution of the width of the railway track from Ostermundigen to Thun for the drive OT_1H. A double gaussian fit has been applied to get the mean value (1436 mm) and the standard deviation (2 mm) of the track width.

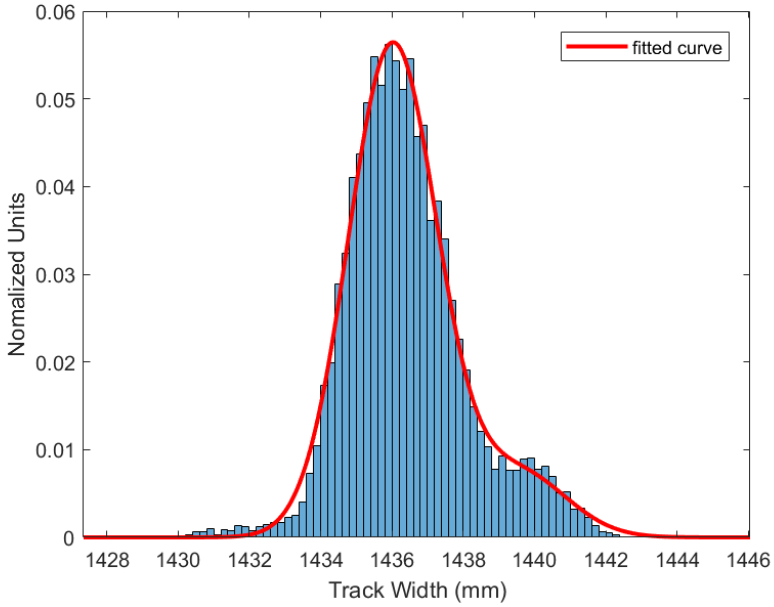

Figure 3-31: The distribution of the measured track width for run OT_1H is shown. A double gaussian fit has been performed to extract the mean value and the standard deviation.

The profile of the track plays also an important role, since the upper part (“Fahrfläche” in Figure 3-30 right) reflects the light more than the internal part. The internal part of the rail is taken as reference for the track width in real world. The upper and more reflecting part is generally more visible in the collected image and it is likely taken as candidate for the line detection alrgorithm.

Figure 3-32 shows the internal (red) and external (green) detected track in the image.

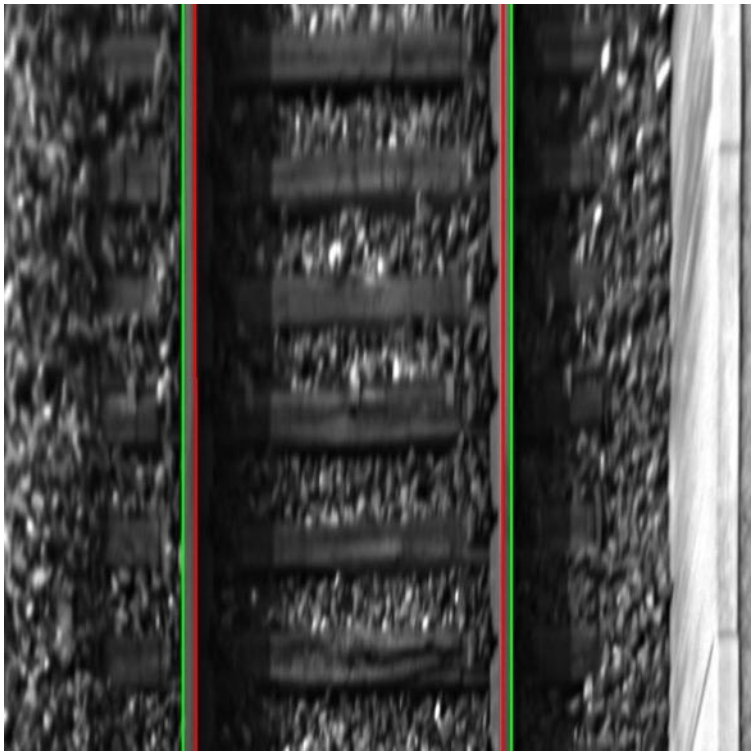

Figure 3-32 The internal (red) and external (green) track profile are identified in the collected image.

The external size of the track can also be used as reference. The “Einbeziehung” is measured with high precision 65.0 ± 0.1 mm, so the relative uncertainty of the track width can be reduced.

The track width is projected to the ground according to the scheme in Figure 3-33. The projected value of the track width is used for the conversion from image pixels to distances in the real world.

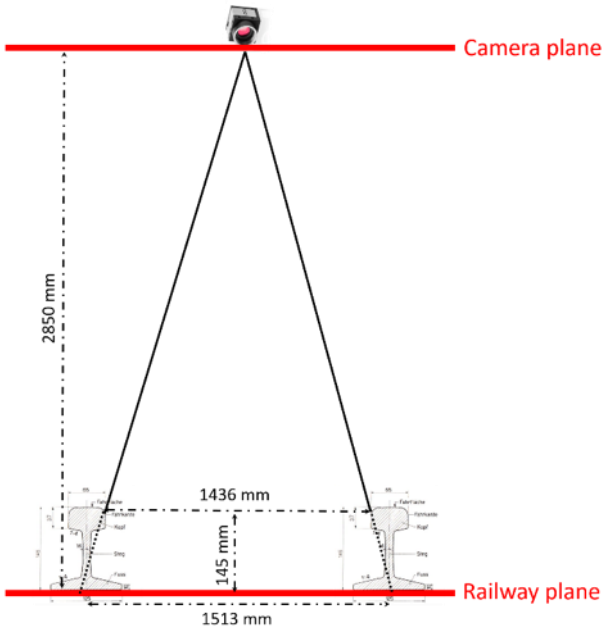

Figure 3-33 The railway track internal width is propagated to the ground, according to the angle of view of a bird-eye image.

3.4.3 Relative localisation with feature tracking

The train position is estimated by means of the optical flow among consecutive images. The optical flow is defined as the pattern of apparent motion of image objects between two consecutive frames caused by the movement of objects or camera.

The sparse optical flow assumes that the pixel intensity of a detected features (strong corner) does not change between consecutive frames. This can lead to uncertainty since the brightness of the environment cannot be controlled and it varies quickly as the train moves (e.g. regions with poor illumination like train stations or tunnels cause a steep increase of the pixel gradient).

The strategy proposed is based on the following steps:

- Feature detection: corners with a brightness gradient are retrieved from the images
- Feature tracking: the optical flow is applied to the detected features
- Calculate pose: the rotation and translation of the camera with respect to the track is calculated for every frame

3.4.3.1 Feature detection

The OpenCV function `goodFeaturesToTrack` [19] is used in order to determine the strong corner in the image. The detected features are then propagated to the next image.

3.4.3.2 Feature tracking

The OpenCV function `calcOpticalFlowPyrLK` [20] is used to calculate the optical flow for the detected features using the iterative Lucas-Kanade method with pyramids [21] in two consecutive images. The output of the function is a list of feature points of the consecutive frames. The points are the coordinates (in pixels) of the features in the two consecutive images. The apparent motion of the features in the images gives a hint of the train motion (Figure 3-34).

In a next step, the results of the optical flow are filtered based on the quality of the tracking.

An additional cut based on the calculated vanishing point of the frame is applied. The calculated tracks from the optical flow shall point to the vanishing point. Tracks outside a defined vanishing point window are filtered out.

Figure 3-34: The features of the image are tracked between consecutive frames. Green circles are the features of the previous frame propagated to the next frame (red circles). The apparent motion of the features is displayed as a red line.

3.4.3.3 Recover pose

Given at least 7 feature points between two images, the rotation and translation of those with respect to the camera can be determined. The OpenCV functions `findEssentialMat` [22] and `recoverPose` [23] are used.

3.4.4 Reduction of drift

The calculation of the driven distance can be improved by freezing the reference frame (key-frame) for a number of following frames, if the estimated velocity is low.

Figure 3-35 shows the 2D position of the train in the drive OT_1H. In red, the position measured from GNSS is shown. This is considered as the reference. In green, the position of the train is calculated based on consecutive frames. In yellow, the position of the train is calculated by means of keyframes. The minimum distance among keyframes is set to 0.5 m. As expected, the keyframe processing improves the accuracy of the estimation of the direction of motion, when the velocity of the train is low. This can be observed in Figure 3-36, where the trajectory without using the key frames (green) drifts away from the reference already after the first curve. The fact that the green curve seems to compensate for the accumulated drift, when the train turns right, is accidental and not associated to any systematic effect in left or right curves.

Figure 3-35: The train position is calculated over the entire path from Ostermundigen to Thun. The position from GNSS (red) is considered as the reference and compared to the train position calculated in consecutve frames (green). In yellow, the position of the train is calculated by means of keyframes.

Figure 3-36: The train position is shown after few kilometes of the drive from Ostermundigen to Thun. The position from GNSS (red) is considered as the reference and compared to the train position calculated in consecutve frames (green). In yellow, the position of the train is calculated by means of keyframes.

3.4.5 Estimation of the Confidence of the result

As described in Section 3.4.2.1, the driven distance is estimated through the template match by minimizing the normalized sum of square differences between the brightness levels of the template and the search images among consecutive frames.

The normalized sum of the square differences has been used as the measure of the quality of the calculated absolute scale. High (low) values of the square difference are likely to be associated to events with wrong (correct) template match.

Figure 3-37 shows the pixel shift of the template image in the drive OT_1H. The pixel shift calculated in frames 8166 and 52803 are not reliable, since the corresponding minimum of the normalized sum of square differences is too high. Such events are filtered out and the average of the preceding five frames is taken instead.

The associated uncertainty to the calculation of the traveled distance is estimated in section 3.6.1.

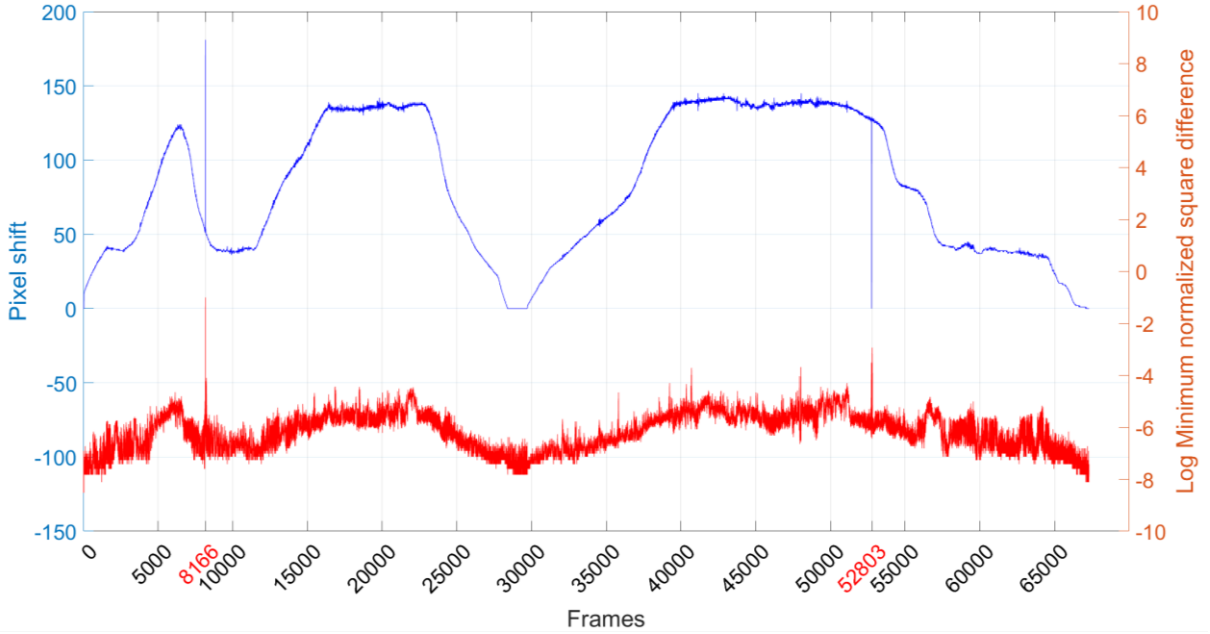

Figure 3-37 The pixel shift of the template in consecutive frames is shown (blue). The quality of the template matching is evaluated by the value of the minimum of the normalized sum of square differences (red). The template matching in frames 8166 and 52803 failed since the minimum of the normalized sum of square difference was too high.

3.4.6 Integration of a simple train model

A Kalman-Filter with a simple train model similar to the one used for FOS described in section 4.5.2 can be used to make the calculated position more robust. Furthermore, the Kalman filter can easily be extended to perform a sensor fusion to prevent the drift of VideoOdometry. Here possible extensions would be to add the detection of AprilTags, points and catenary masts as position updates. The state vector is chosen to be

$$\boldsymbol{x_k} = \begin{bmatrix} x_k \\ y_k \\ v_{x,k} \\ v_{y,k} \\ a_{x,k} \\ a_{y,k} \end{bmatrix}$$

x_k is the x-coordinate of the train position and y_k the y-coordinate of the train position. $v_{x,k}$ and $v_{y,k}$ is the speed in respective direction and $a_{x,k}$ and $a_{y,k}$ the acceleration in respective direction.

The state vector is initialized when the algorithm starts tracking the train and some measurements are already available to estimate all the state values.

A train has limited acceleration and deceleration. Thus, we use a constant acceleration model for the state prediction:

$$\mathrm{x_k} = \mathrm{F_k x_{k-1}} + \mathrm{w_k}$$

with

$$F_k = \begin{bmatrix} 1 & 0 & T & 0 & T^2/2 & 0 \\ 0 & 1 & 0 & T & 0 & T^2/2 \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

the state transition matrix, w_k the process noise and T the sampling time.

Train localisation with video measures the x and y position of the train, so the measurements can be written as

$$z_k = H_k x_k + \nu_k$$

with

$$H_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

the measurement matrix and ν_k the measurement noise.

When you would like to add other measurements to the Kalman-Filter algorithm to improve the results and make it more robust the measurement vector z_k changes. For example, when you add the position updates by AprilTags the measurement vector changes to

$$z_k = \begin{bmatrix} x_{vid} \\ y_{vid} \\ x_{tag} \\ y_{tag} \end{bmatrix}.$$

This also changes the measurement matrix H_k. However, it must be noted that the two sensors have different sample times. Video has a sample frequency of 60Hz, whereas the Tags have no defined sample time. The procedure now is that the Kalman filter works with a frequency of 60Hz whereby the current measurement of the position with video is included in the algorithm with following measurement matrix:

$$H_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

When now a position from the AprilTag is available the measurement matrix changes to

$$H_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

for the next position update with the Kalman Filter.

The procedure to add more sensors is always the same, you only have to consider the sampling time.

Both, the process noise w_k and the measurement noise ν_k are assumed to be zero mean white noise with process covariance Q_k and measurement covariance R_k.

They are defined as the expected values of the process noise and measurement noise vectors:

$$Q_k = E\{w_k w_k^T\} = \begin{bmatrix} \sigma_x & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_y & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{vx} & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{vy} & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{ax} & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{ay} \end{bmatrix}$$

$$R_k = E\{n_k n_k^T\} = \begin{bmatrix} \sigma_{x,vid} & 0 \\ 0 & \sigma_{y,vid} \end{bmatrix}$$

When changing the measurement vector also the covariance matrix for the measurement noise changes. It changes to

$$R_k = E\{n_k n_k^T\} = \begin{bmatrix} \sigma_{x,vid} & 0 & 0 & 0 \\ 0 & \sigma_{y,vid} & 0 & 0 \\ 0 & 0 & \sigma_{x,tag} & 0 \\ 0 & 0 & 0 & \sigma_{y,tag} \end{bmatrix}$$

when adding AprilTags as measurements. The covariance matrix can be used to specify the accuracy of the measurements. Since AprilTags provide an absolute position on the track, the values for $\sigma_{x,tag}$ and $\sigma_{y,tag}$ would be much lower than values $\sigma_{x,vid}$ and $\sigma_{y,vid}$.

This would cause the Kalman filter algorithm to focus more on the position of the AprilTags, if available, than on the position of video. This would correct a possible drift of the position by video.

3.4.7 Optimisation of the parameters

In the following, the impacts of the parameters, which are known to affect the calculation of the train position, has been evaluated.
The parameters are listed in the following:

- Camera calibration (intrinsic parameters)
  Deviation from the calibrated values of the focal length affects the conversion from image to 3D points in the real world. In addition, deviation from the calculated distortion coefficients leads to a lower precision in the railway track detection that is the basis of the camera pose estimation.
- Camera calibration (extrinsic parameters)
  Wrong estimation of the camera extrinsic parameters leads to wrong results in the template match and in the conversion from image to 3D points.
- Standard gauge as reference
  The track width is taken as a reference to calculate the conversion factor from image pixels to distances in the world. The track width is measured by the fit of the distribution. The precision in the knowledge of the vertical distance between the camera mounted on the train and the ground affect the precision of the measured traveled distance.
- Camera height
  The height of the camera was not measured when the camera was mounted on the windscreen and affect the calculation of the absolute scale.
- Track profile
  The track width in the collected image is based on the track detection algorithm, that rely on edge detection. This means, that strong reflecting surfaces emerging from a darker background (or vice versa) are likely candidates for tracks.
  The profile of the track is not regular and introduces an uncertainty in the track position in the image.

In Section 3.6.1, the uncertainties deriving from the above-mentioned parameters on the driven distance have been calculated.

3.5 Object recognition

This section describes the automatic detection of stopping plates and AprilTags by using the tag camera system.

Stopping plates, located at the side of the track, inform the driver about the position the train has to stop. The driver can have an additional benefit from an automatic detection of the stopping plates with the camera system.

AprilTags, mounted on some catenary masts along the side of the track, are object with an exactly known geolocalisation that can be detected by the camera system. First of all, the robustness, accuracy and recognition rate in detecting the AprilTags shall be increased. Then AprilTags can be used as reference to reduce the accumulated drift resulting from the calculation of the train position with Visual Odometry.

3.5.1 Recognition of Stopping Plate (Halteorttafeln) for ATO

The collected video data were analysed in order to detect the “Halteorttafeln“ (“Stopping plate” in the following). The template to be detected is pentagonal with a text in the middle (“H”, “1”, “2” or “3”).

The method described in the following was applied to video data collected with tag and front camera. As expected, better results are obtained using data from the tag camera (Figure 3-38), since a higher focal length is required to clearly identify the template. Furthermore, the tag camera points to the left side of the track, where the stopping plate is supposed to be found (the front camera points to the railway track).

First of all, the collected images are processed for noise reduction and smoothing and adaptive threshold are applied.

Only contours belonging to 5 corners arranged in a given configuration are selected to avoid false-positive candidates. Figure 3-38 shows an identified stopping plate.

This approach can be further improved by analyzing the content of the plate in order to increase the detection efficiency. At the moment, the template is indeed identified using the shape information of the plate only. The analysis can be also extended with the measurement of the position of the plate, that can be possible, once its size is known.

Figure 3-38: The stopping plate identified using the tag camera.

3.5.2 Recognition of AprilTags

In section 3.4, the procedure for a relative train localisation is described. The measured position is expected to drift during the travel. Absolute references like AprilTags are needed to reset the accumulated drift.

For the measurement runs, collected on 14th June 2019, 20 ApriTags, of size of 15.5x15.5 cm^2, were mounted 3 meters height on the catenary masts situated on both sides of the tracks. Ten AprilTags were placed on left side of the track and therefore they were visible to the camera system placed in the locomotive only. Ten AprilTags were placed on right side of the track and therefore they were visible to the camera system placed in the control wagon only.

Figure 3-39 (left) shows the scheme of the camera system moving forward and passing by an AprilTag, that is placed 4 meters on the left side of the track. Figure 3-39 (right) shows one of the collected images. The size of the AprilTag in the image is about 100x100 pixels and the AprilTag is detected.

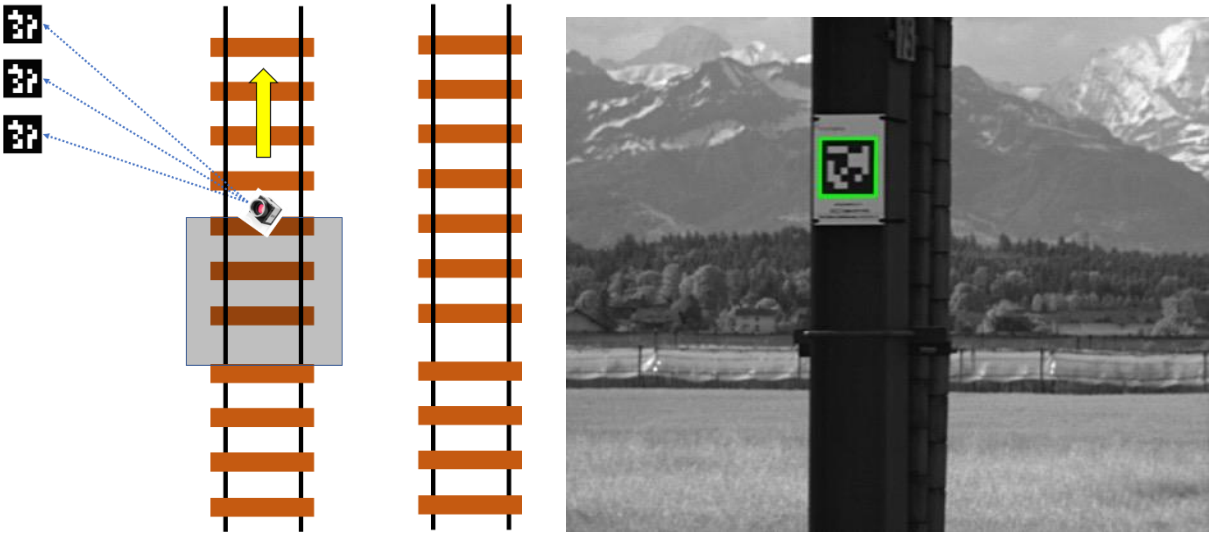

Figure 3-39 Left: scheme of the camera system moving forward and passing by an AprilTag. Right: image where the AprilTag is detected (green square).

Figure 3-40 (left) shows the scheme of the camera system moving backwards and passing by an AprilTag, that is placed about 8 meters on the left side of the track. Figure 3-40 (right) shows one of the collected images. where the AprilTag was detected. The size of the AprilTag in the image is about 60x60 pixels and the AprilTag is detected.

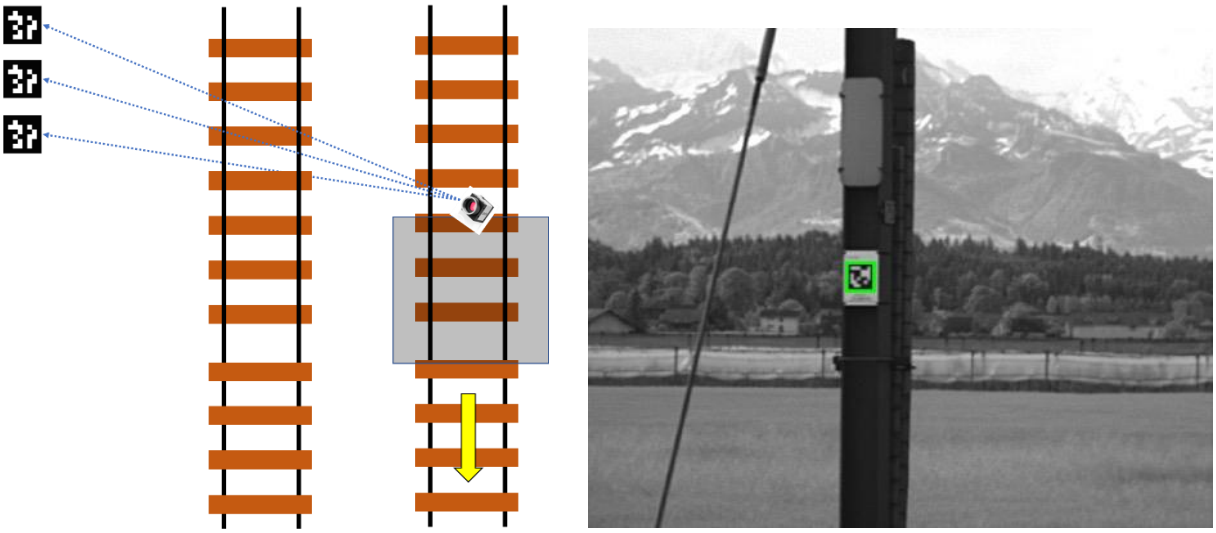

Figure 3-40 Left: scheme of the camera system moving backwards and passing by an AprilTag. Right: image where the AprilTag is detected (green square).

3.6 Results

3.6.1 Measurement of the absolute distance

The different systematic uncertainties are described in Section 3.4.7.

The following parameters (with systematic uncertainties) are estimated for the calculation of the traveled distance:

- Standard gauge from fit of the track width distribution: 1436 ± 2 mm
- Height of the camera mounted on the windscreen: 2850 ± 100 mm
- Detected track distance in pixels: 325 ± 2 pixels
- Camera pose: pitch 8.0 ± 0.1°, yaw 2.5, tilt 2.9
- Focal length: 8.0 ± 0.2 mm

Each parameter is varied within the estimated uncertainty. The difference of the calculated traveled distance with the central value (calculated without including the systematic uncertainty) is shown in Table 3-5. As it can be observed, the dominant contribution is given by the uncertainty deriving from the track detection in the image.

Table 3-5: The traveled distance is calculated by varying each systematic contribution and the difference with respect to the central values is shown.

	Measurement runs							
	OT_1H	**OT_1R**	**OT_2H**	**OT_2R**	**OT_3H**	**OT_3R**	**OT_4H**	**OT_4R**
Uncertainty from fit of track width and from camera height (m)	77	76	74	74	76	76	77	76
Uncertainty from track detection in the image (m)	133	132	128	128	132	132	133	132
Uncertainty from camera pose estimation (m)	127	91	126	84	130	91	131	89
Uncertainty from focal length of the camera (m)	32	32	32	32	32	32	32	32
Central Value (m)	26405	26261	25479	25430	26204	26291	26373	26195

Table 3-6: The total systematic uncertainty of the calculated traveled distance is shown for each measurement run.

	Measurement runs							
	OT_1H	**OT_1R**	**OT_2H**	**OT_2R**	**OT_3H**	**OT_3R**	**OT_4H**	**OT_4R**
Central Value (m)	26405	26261	25479	25430	26204	26291	26373	26195
Total Systematic Uncertainty (m)	202	180	197	173	203	181	204	180
Total Systematic Uncertainty (%)	0.8	0.7	0.8	0.7	0.8	0.7	0.8	0.7

The total systematic uncertainty of the measured travelled distance ranges from 0.7 to 0.8%, as it can be seen in Table 3-6.

The comparison of the results with the GNSS (combined with IMU) and GTG references is described in section 6.3.1.

3.6.2 Measurement of the local position

In order to track the train motion in the ground reference system, the following information should be available:

- Position of the starting points (Ostermundigen and Thun train stations in this case):
  - This information is taken from GNSS:
- Initial yaw of the train:
  - This information is taken from track topography (GTG)
- Camera extrinsic parameters:
  - Measured as explained in Section 3.3.2
- Absolute scale:
  - Measured as explained in Section 3.4.2

The results of the calculated train position are shown in Section 6.3.3 and compared to the GNSS/IMU combination.

3.6.3 Detection of the AprilTags

The results of the detection of the AprilTags during the travel from Ostermundigen to Thun are summarized in the following tables.

The tables contain the following data:

- Column *ID*: Identification number of the detected AprilTags
- Column *Lat*: Nominal latitude of the AprilTags
- Column *Lon*: Nominal longitude of the AprilTags
- Column *Passing Time*: Time, when the train passes the AprilTag. This is calculated from the tag position, which is calculated in every frame where the tag is detected, and it is back-propagated to an ideal 0-distance (distance where the train passes the tag) using the travelled distance calculated with the template match in images, that have been collected from the front camera. The uncertainties (in parenthesis) are both statistical and systematics and refer to the last digits. The statistical error is derived from the different position of the detected tags in consecutive frames. The standard deviation of the arrival times is the statistical error. The systematic uncertainty takes into account the image processing required to identify the tag, the method used to recover the tag pose and the tag size. The uncertainty due to the propagation using the template match has been found to be negligible.
- Column *Uncertainty*: The statistical uncertainty of the arrival time is combined to the systematic uncertainty and converted to spatial resolution.
- Column *Nr frames*: The number of frames where the AprilTag is detected.

Table 3-7: The results of the tag detection during the drive from Ostermundigen to Thun (OT_1H) are listed. The tag with ID 4 was detected in few frames with a dedicated image processing. The statistical and systematic error are therefore not accurate.

Id	Lat	Lon	Passing time (Stat)(Sys)	Uncertainty (m)	Nr frames
28	46,8695	7,5606	06:53:58.973 (002)(024)	0.845	10
13	46,8678	7,5610	06:54:04.008 (001)(038)	1.426	4
4	46,8668	7,5613	06:54:07.080 (000)(001)	0.012	4
20	46,8648	7,5619	06:54:13.248 (002)(017)	0.639	9
18	46,8639	7,5622	06:54:15.931 (009)(021)	0.864	9
29	46,8382	7,5698	06:55:36.145 (002)(020)	0.763	9
17	46,8372	7,5701	06:55:39.151 (008)(029)	1.130	10
23	46,8362	7,5704	06:55:41.974 (015)(024)	1.093	9
8	46,8352	7,5707	06:55:44.950 (015)(025)	1.155	9
3	46,8337	7,5711	06:55:49.425 (002)(021)	0.882	10

Table 3-8: the results of the tag detection during the drive from Thun to Ostermundigen (OT_1R) are listed. The tag with ID 4 was not be detected while the tag with ID 28 was not visible due to another train passing by on the other track.

Id	Lat	Lon	Passing time (Stat)(Sys)	Uncertainty (m)	Nr frames
3	46,8337	7,5711	07:16:40.011 (007)(043)	1.653	19
8	46,8352	7,5707	07:16:44.574 (002)(043)	1.642	18
23	46,8362	7,5704	07:16:47.597 (007)(044)	1.689	19
17	46,8372	7,5701	07:16:50.474 (005)(046)	1.797	20
29	46,8382	7,5698	07:16:53.495 (006)(047)	1.818	19
18	46,8639	7,5622	07:18:08.628 (008)(044)	1.692	18
20	46,8648	7,5619	07:18:11.256 (003)(046)	1.725	19
4	46,8668	7,5613	Not detected		
13	46,8678	7,5610	07:18:20.476 (011)(048)	1.841	7
28	46,8695	7,5606	Occluded by other train		

Table 3-9: The results of the tag detection during the drive from Ostermundigen to Thun (OT_2H) are listed.

Id	Lat	Lon	Passing time (Stat)(Sys)	Uncertainty (m)	Nr frames
28	46,8695	7,5606	08:10:25.953 (013)(111)	0.828	46
13	46,8678	7,5610	08:10:48.735 (006)(098)	0.855	41
4	46,8668	7,5613	08:11:01.408 (012)(092)	0.881	37
20	46,8648	7,5619	08:11:25.170 (003)(086)	0.839	35
18	46,8639	7,5622	08:11:35.052 (005)(081)	0.836	34
29	46,8382	7,5698	08:13:26.125 (015)(024)	1.010	10
17	46,8372	7,5701	08:13:29.314 (014)(026)	1.069	10
23	46,8362	7,5704	08:13:32.309 (009)(022)	0.872	10
8	46,8352	7,5707	08:13:35.450 (002)(024)	0.899	10
3	46,8337	7,5711	08:13:40.155 (002)(018)	0.669	10

Table 3-10: The results of the tag detection during the drive from Thun to Ostermundigen (OT_2R) are listed.

Id	Lat	Lon	Passing time (Stat)(Sys)	Uncertainty (m)	Nr frames
3	46,8337	7,5711	08:33:41.947 (015)(044)	1.753	19
8	46,8352	7,5707	08:33:46.551 (013)(043)	1.694	19
23	46,8362	7,5704	08:33:49.588 (012)(045)	1.788	19
17	46,8372	7,5701	08:33:52.448 (018)(048)	1.963	20
29	46,8382	7,5698	08:33:55.479 (014)(047)	1.880	20
18	46,8639	7,5622	08:35:37.249 (020)(112)	1.660	49
20	46,8648	7,5619	08:35:44.324 (018)(127)	1.666	56
4	46,8668	7,5613	08:36:03.249 (030)(150)	1.797	60
13	46,8678	7,5610	08:36:13.010 (022)(141)	1.717	59
28	46,8695	7,5606	08:36:28.665 (030)(156)	1.787	57

Table 3-11: The results of the tag detection during the drive from Ostermundigen to Thun (OT_3H) are listed.

Id	Lat	Lon	Passing time (Stat)(Sys)	Uncertainty (m)	Nr frames
28	46,8695	7,5606	09:26:08.653 (009)(069)	0.838	28
13	46,8678	7,5610	09:26:22.726 (014)(061)	0.877	25
4	46,8668	7,5613	09:26:30.567 (013)(060)	0.943	23
20	46,8648	7,5619	09:26:44.758 (011)(050)	0.868	21
18	46,8639	7,5622	09:26:50.462 (016)(049)	0.921	19
29	46,8382	7,5698	09:28:27.132 (009)(023)	0.926	9
17	46,8372	7,5701	09:28:30.172 (014)(026)	1.186	9
23	46,8362	7,5704	09:28:33.046 (012)(022)	0.958	9
8	46,8352	7,5707	09:28:36.056 (002)(025)	0.641	9
3	46,8337	7,5711	09:28:40.578 (009)(022)	0.922	9

Table 3-12: The results of the tag detection during the drive from Thun to Ostermundigen (OT_3R) are listed. The tag with ID 13 was not visible due to another train passing by on the other track.

Id	Lat	Lon	Passing time (Stat)(Sys)	Uncertainty (m)	Nr frames
3	46,8337	7,5711	10:00:54.920 (016)(047)	1.791	20
8	46,8352	7,5707	10:00:59.717 (011)(046)	1.720	20
23	46,8362	7,5704	10:01:02.983 (017)(049)	1.856	20
17	46,8372	7,5701	10:01:06.067 (008)(052)	1.919	22
29	46,8382	7,5698	10:01:09.328 (010)(051)	1.862	20
18	46,8639	7,5622	10:02:47.821 (016)(100)	1.699	43
20	46,8648	7,5619	10:02:53.963 (021)(107)	1.717	45
4	46,8668	7,5613	10:03:08.975 (030)(121)	1.736	41
13	46,8678	7,5610	Occluded by other train		
28	46,8695	7,5606	10:03:36.567 (031)(193)	1.850	81

Table 3-13: The results of the tag detection during the drive from Ostermundigen to Thun (OT_4H) are listed.

Id	Lat	Lon	Passing time (Stat)(Sys)	Uncertainty (m)	Nr frames
28	46,8695	7,5606	11:38:17.013 (014)(039)	0.937	15
13	46,8678	7,5610	11:38:25.201 (015)(048)	1.150	10
4	46,8668	7,5613	11:38:30.336 (000)(039)	0.888	2
20	46,8648	7,5619	11:38:40.870 (003)(038)	0.823	16
18	46,8639	7,5622	11:38:45.487 (002)(043)	0.926	16
29	46,8382	7,5698	11:41:00.482 (007)(038)	0.791	17
17	46,8372	7,5701	11:41:06.622 (006)(046)	0.837	18
23	46,8362	7,5704	11:41:13.094 (009)(052)	0.847	21
8	46,8352	7,5707	11:41:21.234 (004)(062)	0.826	26
3	46,8337	7,5711	11:41:36.333 (014)(078)	0.863	30

Table 3-14: The results of the tag detection during the drive from Thun to Ostermundigen (OT_4R) are listed. The tag with ID 3 was not visible due to another train passing by on the other track.

Id	Lat	Lon	Passing time (Stat)(Sys)	Uncertainty (m)	Nr frames
3	46,8337	7,5711	Occluded by other train		
8	46,8352	7,5707	12:18:52.568 (011)(073)	1.672	31
23	46,8362	7,5704	12:18:57.679 (006)(077)	1.735	34
17	46,8372	7,5701	12:19:02.547 (011)(082)	1.863	34
29	46,8382	7,5698	12:19:07.688 (004)(080)	1.808	33
18	46,8639	7,5622	12:21:16.877 (018)(075)	1.721	32
20	46,8648	7,5619	12:21:21.394 (011)(076)	1.683	32
4	46,8668	7,5613	12:21:31.747 (022)(077)	1.781	28
13	46,8678	7,5610	12:21:36.984 (014)(075)	1.683	32
28	46,8695	7,5606	12:21:45.371 (020)(083)	1.864	34

As it can be seen in Table 3-7 to Table 3-14, the uncertainty in the measured position is higher when the train moves backwards, in runs OT_1R, OT_2R, OT_3R and OT_4R. This is due to the fact that the AprilTags are placed on the left side of the track. Indeed, the AprilTags are closer to the camera when the train moves forward.

The detection efficiency can be defined as the number of AprilTags detected, divided by the total number of AprilTags that were visible from the camera. In this way, AprilTags occluded by a train passing in the other tracks do not count.

Table 3-15 shows the efficiency of the detection of AprilTags for each run. The overall detection efficiency is 100% when the train move forward and 97% when the train move backward.

Table 3-15: The efficiency of the detection of AprilTags.

Run	Train Direction	ApriTags			Detection efficiency
		visible	occluded	detected	
OT_1H	forward	10	0	10	100%
OT_1R	backward	9	1	8	89%
OT_2H	forward	10	0	10	100%
OT_2R	backward	10	0	10	100%
OT_3H	forward	10	0	10	100%
OT_3R	backward	9	1	9	100%
OT_4H	forward	10	0	10	100%
OT_4R	backward	9	1	9	100%

3.6.4 Measurement of the extrinsic parameters with a train at rest

On 3rd December 2019 video data were collected on a RE420. The measurement required a train at rest for the estimation of the camera pose.
The camera box has been mounted in different positions at the windscreen, pointing the railway track. For each camera pose, video data containing 100 frames were stored for post-processing analysis:

- video_front_ctrl_191203_1428_frame_0.raw
- video_front_ctrl_191203_1431_frame_0.raw
- video_front_ctrl_191203_1432_frame_0.raw (not included since the camera yaw angle changes)
- video_front_ctrl_191203_1433_frame_0.raw
- video_front_ctrl_191203_1434_frame_0.raw
- video_front_ctrl_191203_1442_frame_0.raw
- video_front_ctrl_191203_1449_frame_0.raw
- video_front_ctrl_191203_1452_frame_0.raw
- video_front_ctrl_191203_1456_frame_0.raw
- video_front_ctrl_191203_1458_frame_0.raw
- video_front_ctrl_191203_1500_frame_0.raw
- video_front_ctrl_191203_1503_frame_0.raw

The procedure for the automatic estimation of the camera pose is described in Section 3.3.2. In the following, a summary of the results and detailed plots are shown for each dataset.

3.6.4.1 Summary

The railway track detection efficiency is measured and compared for each dataset in Figure 3-41.
The lower detection efficiency in dataset 1456 is due to the fact that the camera was not well posed. The camera pitch is too small meaning that the camera points to a region outside the camera focus. This results in a poor quality of the image acquired. This can be seen in the detailed description of the result for that dataset.
Lower detection efficiency has been observed in dataset 1442 and is under investigation. It could be due to the dirty condition of the windscreen, as can be seen in the detailed description of the result of that dataset.

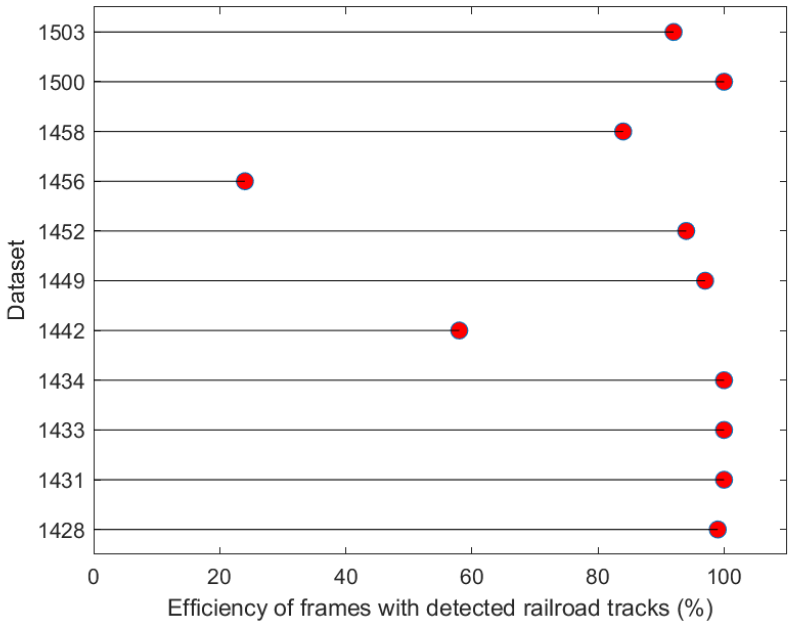

Figure 3-41 Summary of the percentage of the frames where the track was detected.

Figure 3-42 shows the pitch and yaw calculated in each dataset. The red band is the standard deviation of the angle distribution.

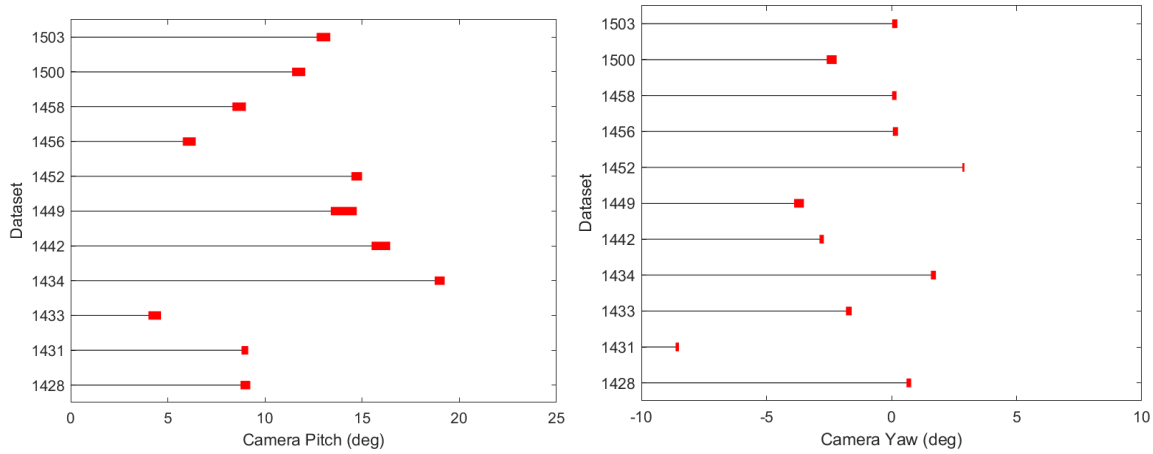

Figure 3-42 (left) Summary of the estimated pitch, (right) Summary of the estimated yaw.

It can be seen that the precision in the estimation of the yaw is higher with respect to the one of the pitch.

3.6.4.2 Datasets

In the following, the distributions of the pitch and yaw of each datasets are shown. The lower resolution measured in some of the datasets is related to the poor measurement of the position of the railway track in the image.

3.6.4.2.1 Dataset 1428

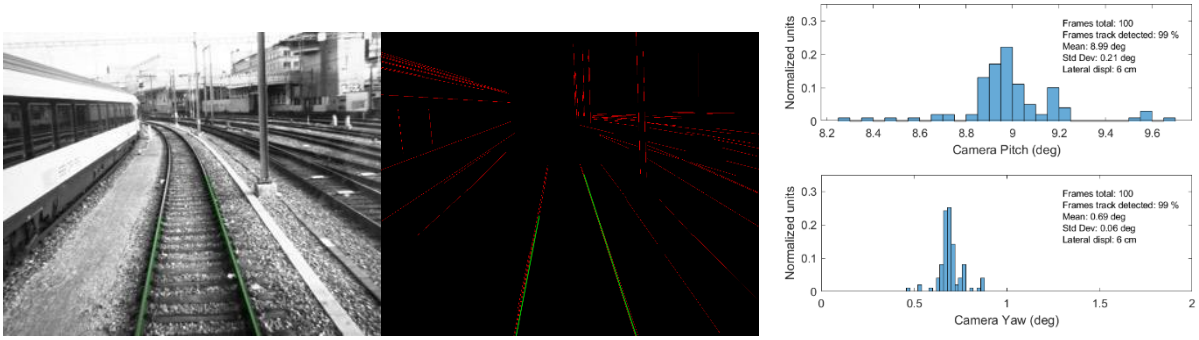

Figure 3-43 Left: Image collected from the camera with identified railway track (green). Middle: Image processed with detected lines (red) and identified railway track (green). Right: Estimated pitch (top) and yaw (bottom) distributions.

3.6.4.2.2 Dataset 1431

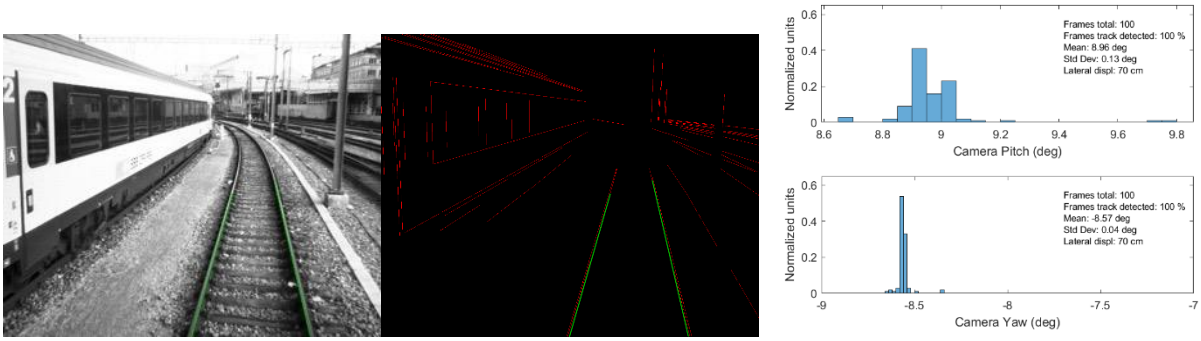

Figure 3-44 Left: Image collected from the camera with identified railway track (green). Middle: Image processed with detected lines (red) and identified railway track (green). Right: Estimated pitch (top) and yaw (bottom) distributions.

3.6.4.2.3 Dataset 1433

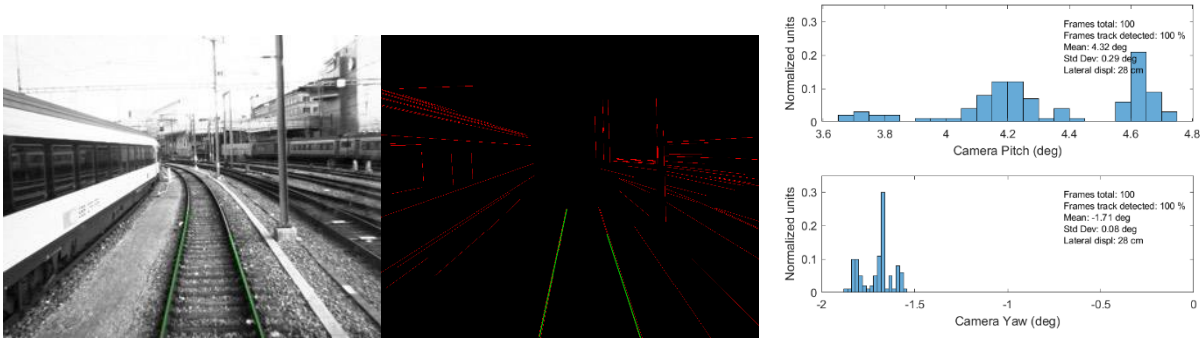

Figure 3-45: Left: Image collected from the camera with identified railway track (green). Middle: Image processed with detected lines (red) and identified railway track (green). Right: Estimated pitch (top) and yaw (bottom) distributions.

3.6.4.2.4 Dataset 1434

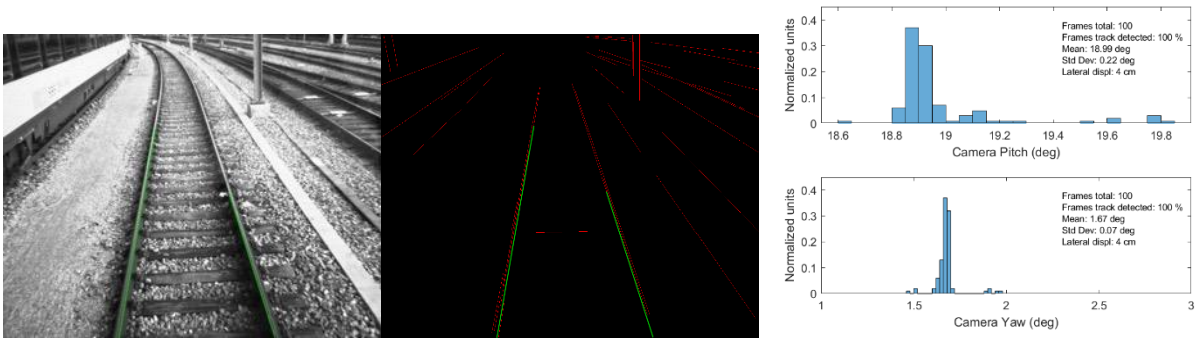

Figure 3-46: Left: Image collected from the camera with identified railway track (green). Middle: Image processed with detected lines (red) and identified railway track (green). Right: Estimated pitch (top) and yaw (bottom) distributions.

3.6.4.2.5 Dataset 1442

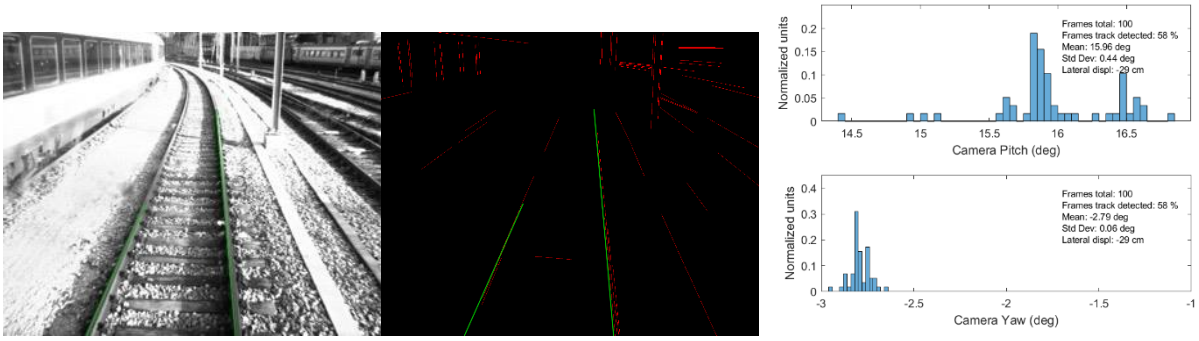

Figure 3-47: Left: Image collected from the camera with identified railway track (green). Middle: Image processed with detected lines (red) and identified railway track (green). Right: Estimated pitch (top) and yaw (bottom) distributions.

3.6.4.2.6 Dataset 1449

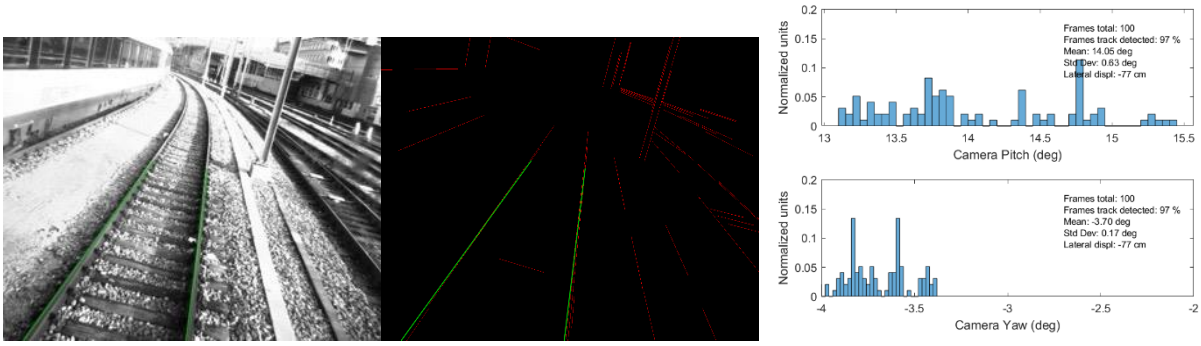

Figure 3-48: Left: Image collected from the camera with identified railway track (green). Middle: Image processed with detected lines (red) and identified railway track (green). Right: Estimated pitch (top) and yaw (bottom) distributions.

3.6.4.2.7 Dataset 1452

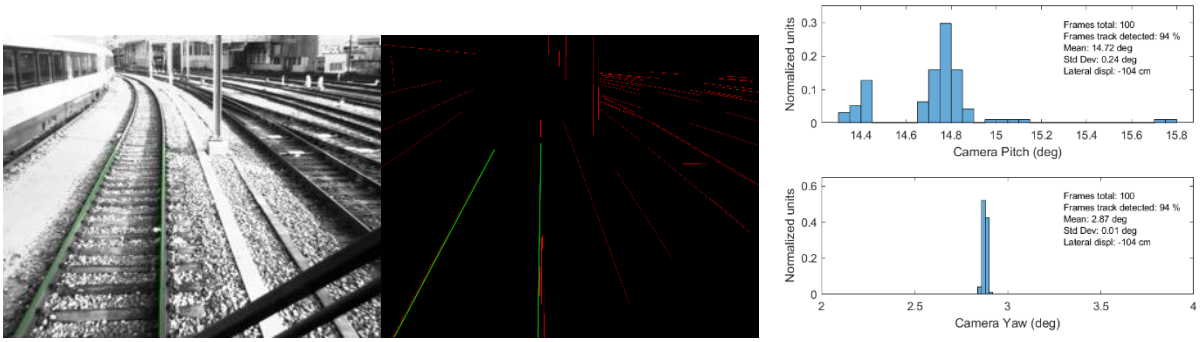

Figure 3-49: Left: Image collected from the camera with identified railway track (green). Middle: Image processed with detected lines (red) and identified railway track (green). Right: Estimated pitch (top) and yaw (bottom) distributions.

3.6.4.2.8 Dataset 1456

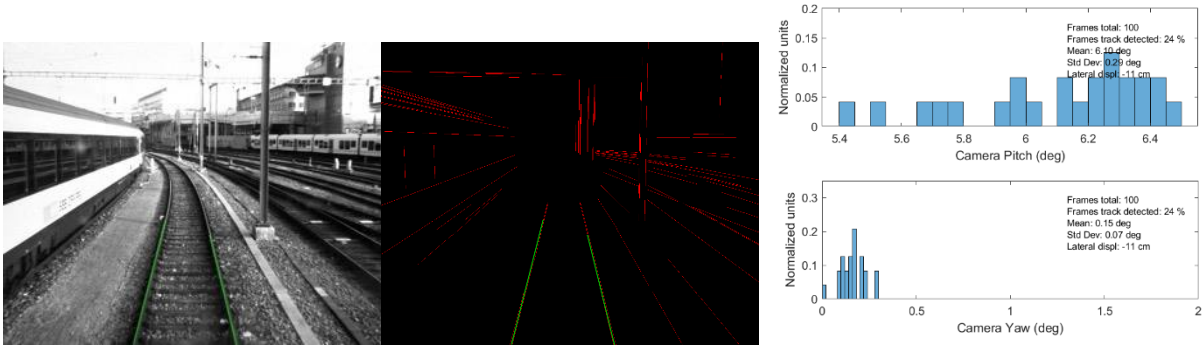

Figure 3-50: Left: Image collected from the camera with identified railway track (green). Middle: Image processed with detected lines (red) and identified railway track (green). Right: Estimated pitch (top) and yaw (bottom) distributions.

3.6.4.2.9 Dataset 1458

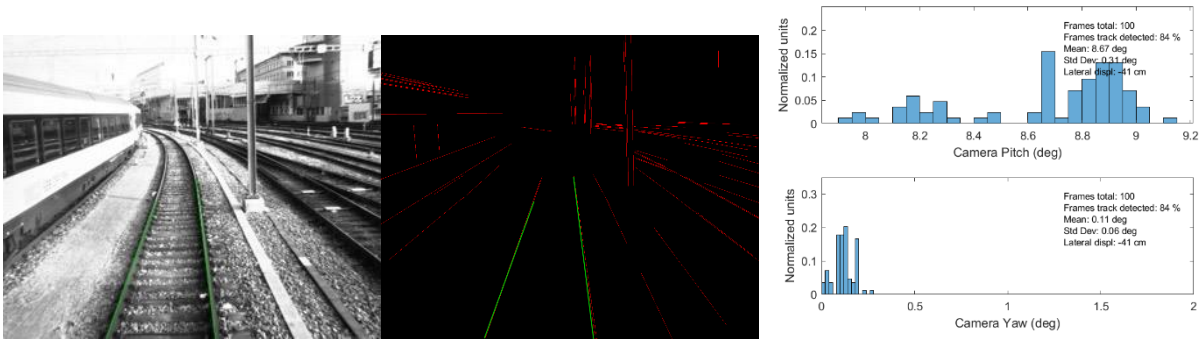

Figure 3-51: Left: Image collected from the camera with identified railway track (green). Middle: Image processed with detected lines (red) and identified railway track (green). Right: Estimated pitch (top) and yaw (bottom) distributions.

3.6.4.2.10 Dataset 1500

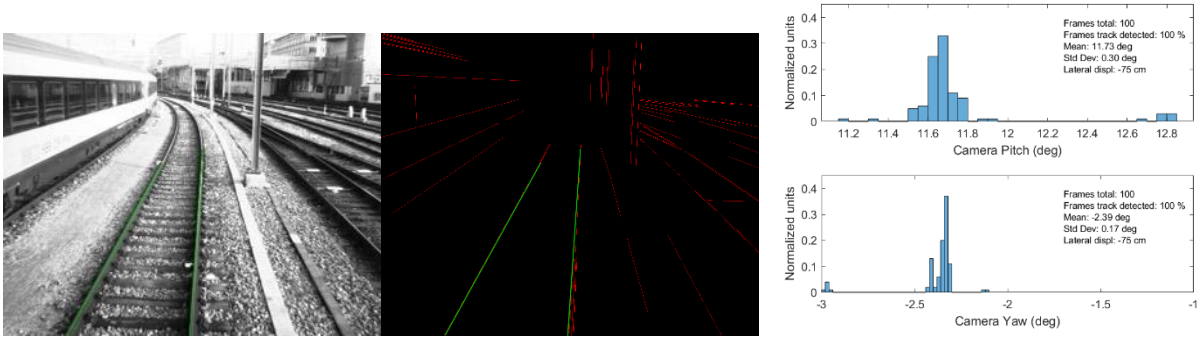

Figure 3-52: Left: Image collected from the camera with identified railway track (green). Middle: Image processed with detected lines (red) and identified railway track (green). Right: Estimated pitch (top) and yaw (bottom) distributions.

3.6.4.2.11 Dataset 1503

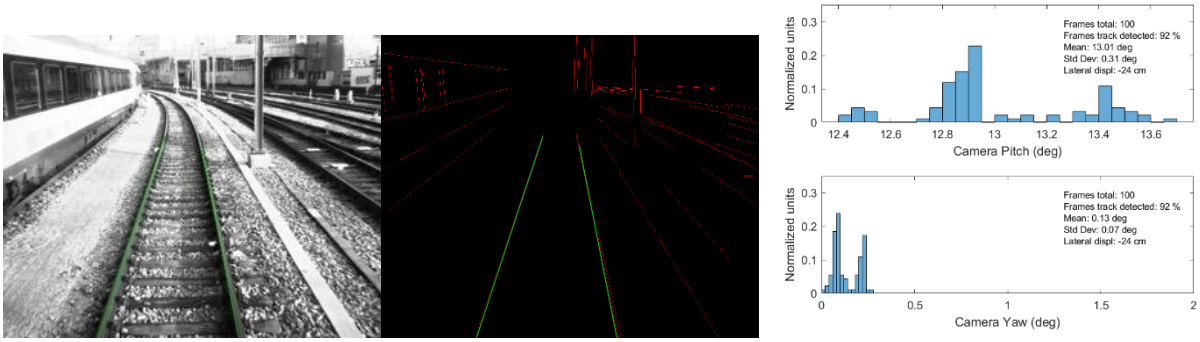

Figure 3-53: Left: Image collected from the camera with identified railway track (green). Middle: Image processed with detected lines (red) and identified railway track (green). Right: Estimated pitch (top) and yaw (bottom) distributions.

3.6.5 Measurement at night / darkness

The camera box is equipped with infrared illuminators in order to enlighten dark scenes like tunnels. The camera box is placed within the train behind the windscreen. This limits the power of the illuminator since only part of the light emitted can pass though the screen and then illuminate the scene. As mentioned in section 3.2.4, in condition with poor illumination, the camera exposure and the camera gain are set to the maximum allowed values.

In Figure 3-54 the collected images are rectified and then transformed to get a bird-eye view. The left picture shows the image scaled from 10 to 8 bits linearly. In the center, an image scaled from 10 to 8

bits logarithmically is shown. By scaling according to a logarithmic function, the brightness difference of dark pixels is enhanced with respect to bright pixels. The right picture shows the result of a threshold-operation according to the mean brightness in a region of interest containing the track. The track lines, pointed from the red arrows in Figure 3-54 (right), can be hardly seen.

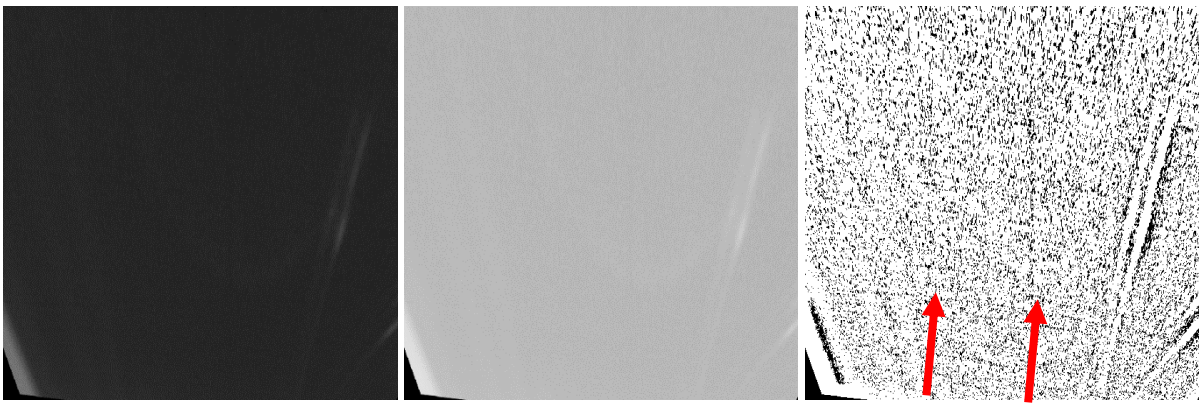

Figure 3-54: Left: The collected image is converted from 10 to 8 bits linearly. Middle: The collected image is converted from 10 to 8 bits logarithmically. Right: The logarithmically converted image is processed with an adaptive threshold in order to detect the railway tracks. The track lines, pointed from the red arrows, are poorly visible in the image and cannot be detected.

In the example, the calculation of the absolute distance and the tracking of features is not possible. A possible solution is to increase the illumination of the scene by increasing the power of the illuminator or by placing it outside the train, so that part of the light will not be reflected from the screen.

3.6.6 Measurement with snow

The performance of the railway track recognition and the determination of the train speed has been tested on measurements with snow.

On snowy days, both camera exposure and gain are increased in order to compensate for the lack of illumination. In addition, the snow laying on the track can be very inhomogeneous, meaning that the brightness levels of the region of interest in front of the track can vary a lot between consecutive frames. This means that the gain and exposure times shall be accurately controlled in order to avoid artificial jumps in the brightness of the collected image frames, that are not due to brightness changes in the scene. Moreover, using the mean value of the brightness for the automatic control function could be not the best choice since a dark scene with snow spots would lead to an acceptable average brightness level, although the scene could be not well exposed.

As described in section 3.4.2, the calculation of the absolute distance shouldn’t be influenced by the change of brightness in consecutive frame. Indeed, the template matching is based on the minimization of the brightness square difference normalized to the brightness of the images in use.

Figure 3-55 shows the image collected with snow on the track (left) and the railway track can be identified (center). As expected, the template matching is not affected by the snow on the track and the speed can be measured (right).

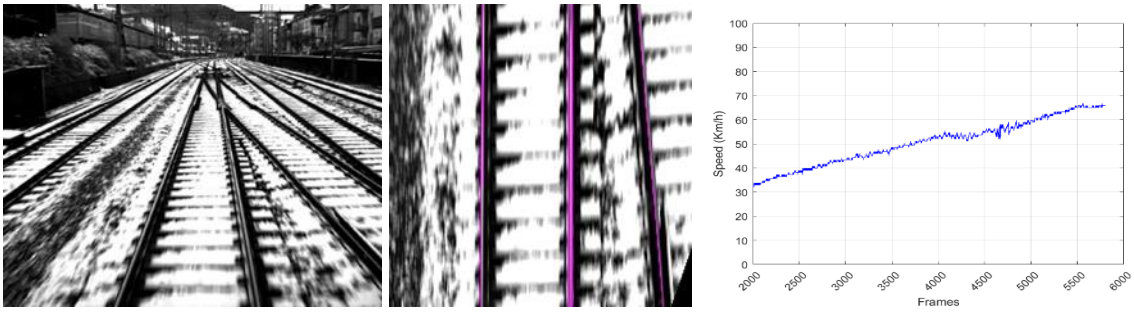

Figure 3-55 (left): Image collected from St Gallen to Bern (data from drive BSG_1R) with snow on the railway track, (center): the image is transformed to a bird-eye view where the railway track (magenta) can be identified, (right): the absolute speed is measured with precision meaning that the template matching is not affected by the snow.

Figure 3-56 shows the image collected with snow on the track (left). As it can be observed, the image is blurry since the exposure is set to the maximum value due to the lack of illumination and the train speed is high. The railway track is difficult to identify, as can be seen in Figure 3-56 (center). As expected, the template matching is not affected by the blurry image and the speed can be measured (right).

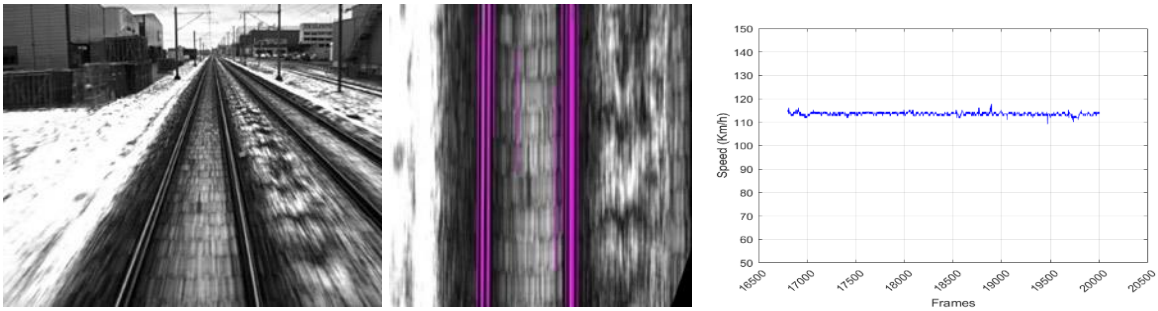

Figure 3-56: (left) Image collected from St Gallen to Bern with little snow on the railway track. Motion blur can be observed due to low illumination of the scene and high speed of the train, (center) the image transformed to a bird-eye view where the railway track (magenta) is difficult to be identified, (right): the absolute speed is measured with precision meaning that the template matching is not affected by the snow and by the motion blur.

3.6.7 Measurement with heavy rain

Unfortunately, no heavy rain was present in any of the measurements.

3.6.8 Measurement on a foggy day

Unfortunately, no fog was present in any of the measurements.

3.6.9 Measurement on sunny day with shadows

As explained in section 3.2.4, the camera exposure is set depending on the average brightness levels of a region in front of the train. The scene is well exposed if the brightness levels of the scene are homogeneous. In case of a bright scene with shadows, the brightness picks both a low and high value.

Figure 3-57 (left) shows the shadow on the railway track caused by a train passing alongside. The railway track can be detected (green, Figure 3-57 center) and the train speed can be calculated, meaning the travelled distance by means of the template matching is not affected by the shadows in the example.

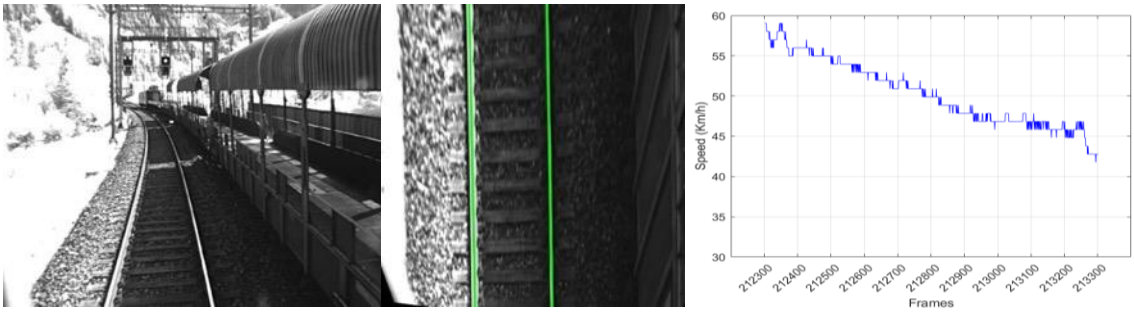

Figure 3-57 (left) Image collected from the camera, where a train is passing nearby causing shadow on the track, (center) the image transformed to a bird-eye view where the railway track (green) can be identified, (right): the absolute speed is measured with precision meaning that the template matching is not affected by the shadows.

Figure 3-58 (left) shows the shadow on the railway track caused by a small tunnel. The railway track can be detected (green, Figure 3-58 center) and the train speed can be calculated, meaning the travelled distance by means of the template matching is not affected by the shadows in the example.

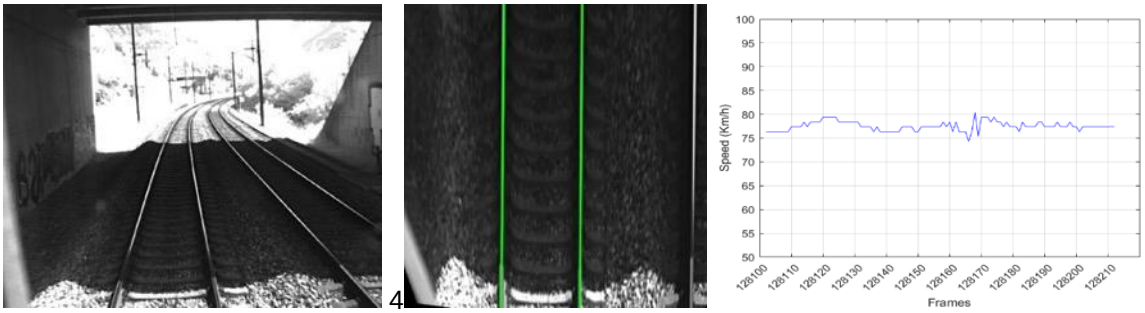

Figure 3-58 (left) Image collected from the camera, where a train is passing a short tunnel, (center) the image transformed to a bird-eye view where the railway track (green) can be identified, (right): the absolute speed is measured with precision meaning that the template matching is not affected by the shadow of the short tunnel.

3.6.10 Measurement on bridges

Figure 3-59 (left) shows the image collected from the train passing the Uttigen bridge. The railway track can be detected (green, Figure 3-59 center) and the train speed can be calculated, meaning the travelled distance by means of the template matching is not affected by structure of the railway track in the example.

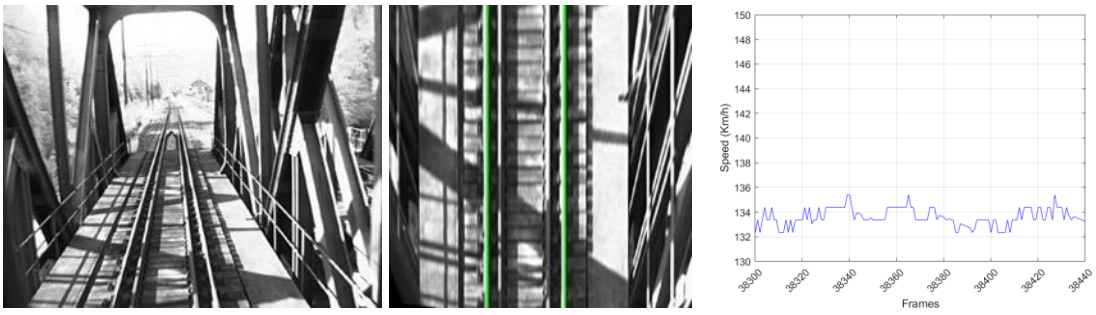

Figure 3-59 (left) Image collected from the camera, where a train is passing the Uttigen bridge, (center) the image transformed to a bird-eye view where the railway track (green) can be identified, (right): the absolute speed is measured with precision meaning that the template matching is not affected by the ground structure of the bridge.

Figure 3-60 (left) shows the image collected from the train passing a bridge close to Brig. The railway track can be detected (green, Figure 3-60 center) and the train speed can be calculated, meaning the travelled distance by means of the template matching is not affected by structure of the railway track in the example.

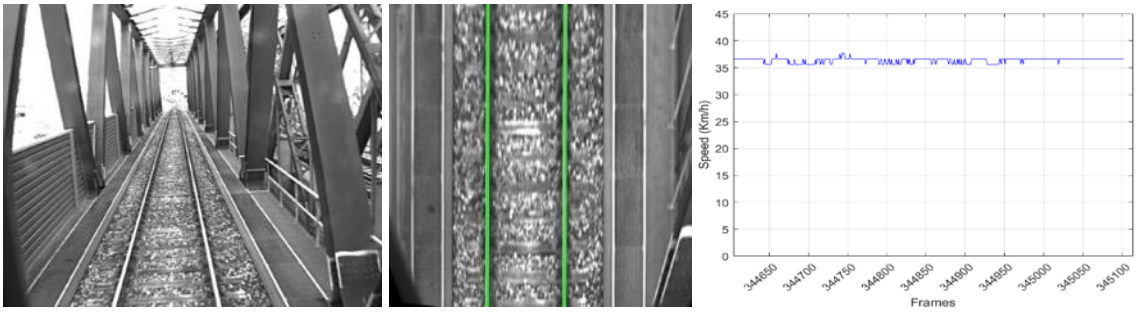

Figure 3-60 (left) Image collected from the camera, where a train is passing the bridge before the Brig station, (center) the image transformed to a bird-eye view where the railway track (green) can be identified, (right): the absolute speed is measured with precision meaning that the template matching is not affected by the ground structure of the bridge.

3.6.11 Measurement with a tilting train

Unfortunately, no measurement with a tilting functionality was available. The tilting functionality on RABDe 500 did not work on the day of the measurement.

4 Sensor Technology FOS (Fiber Optic Sensing)

It is customary for communications companies to install their fiber optic cables running alongside train tracks for various reasons. Many pairs of cables are usually installed, resulting in spare cables that can be used for expansion or other purposes.

FOS can use a pair of fiber optic cables for runs of currently up to 40 Km in order to sense mechanical vibrations (sound) in sections of the fiber optic cable creating a high sensitivity passive distributed sensor which can report the vibration profile across the whole cable at a rate of thousands of times per second.

4.1 Objectives

The objective of using FOS as a sensor is to be able to track moving trains, their position, front and rear ends, velocity, length, etc., in almost real time, with a predefined maximum allowed latency defined by the user.

As the name implies, FOS works by using an already existing fiber optic cable which was already laid out alongside the train tracks.

As long as the train is moving above a certain speed, FOS can pinpoint its location with absolute accuracy and is not subject to error accumulation. It also works well in tunnels, where some technologies are not available (GNSS) or have great difficulty dealing with the low light conditions (Video).

4.2 Introduction

The fiber optic cable used for the FOS measurements was installed adjacent to the train tracks in an almost straight line from Münsingen to Thun as depicted in Figure 4-1. For the purposes of acoustic sensing, the fiber was logically divided into 1190 segments of 8.167619 meters in length each and the measurements were done using the ODH-3 equipment from OptaSense.

Figure 4-1: Fiber optic cable path

The fiber optic cable used is an ordinary cable which was already installed for communications purposes and no special installation was used for FOS expect for the active components at the endpoints. A calibration run was done by SBB and Optasense in order to map each catenary mast along the track to its respective fiber channel. From there it was possible to interpolate each channel to its linear position along the track, which is essential to determine the physical quantities we are looking for (see Table 4-1 much further below).

In broad terms, each section of a fiber optic cable (channel) is represented by a certain point in it whose relative displacement in relation to its resting position is measured at predefined intervals (sampling rate). This displacement varies greatly from channel to channel and is dependent not only on the source signal but also on many factors which attenuate and distort the signal until it reaches this measuring point.

It is expected that the measurement values of each channel are proportional to the acoustic vibration (sound) at its representative point and the whole fiber optic cable can be seen as a sequence of independent microphones. In this model, however, each microphone has a different attenuation profile and reports signals which, in general, bear little resemblance with each other, especially in relation to their dynamic ranges, even though we can expect “gradual” variations from channel to channel.

In order to present the results and algorithms, we have chosen an interval of data where the measuring train was present, along with some other trains that usually ride on this track segment.

The chosen interval is a 15-minute interval from 14-June-2019, starting 0.2 ms after 6:45 am GMT which, for all practical purposes, can be regarded at 6:45 am sharp (2019-06-14T06:45:00:000200Z).

The data is composed of 32-bit (4-byte) integers sampled at 2500 Hz for each of the 1190 channels. The total date rate for this segment is, therefore, 4 x 2500 x 1190 = 11.900.000 bytes per second to be processed in almost real time.

4.3 Analysis Architecture

The FOS analysis can be divided into 2 clear parts:

1. **Intra channel**: which looks at each channel independently of all others. Intra channel analysis should be responsible for the filtering, thresholding, power and spectra calculation, and any other transformations done behaving as each channel is independent of all others.
2. **Inter channel**: A higher level which only deals with the events generated from the intra channel analysis and which looks across all channels, actually tracking the moving objects. This level should also deal with the physics of the tracked object and, ideally, should feedback its results or predictions to the intra channel level which could use this information in order to dynamically adjust its running parameters, increasing both the accuracy and the precision of the whole system.

The intra channel analysis has the goal of generating discrete events for each unit of time, e.g., “train on” and “train off” events when there is a train passing or not, respectively, in front of this channel. These events are then used by the inter channel analysis to track the train front and rear ends across channels in relation to time. The inter channel analysis can then be used to report train position, speed, and length among other values.

4.3.1 Signal and Silence

It is extremely important to be able to detect “silence” periods, i.e., periods of time during which “no” vibration or sound is being reported. In other words, it is imperative to be able to detect and classify background noise and differentiate it from periods when a signal is and is not present. If silence periods can be reliably detected, any deviation from it is a noteworthy event that should be investigated

Unfortunately, as will be discussed later, the signal which gets reported when there is no vibration does not possess the characteristics of neither white nor pink noise and presents some very high-power low frequency components.

Apart from thermal and systemic noises, it must be pointed out that there can be other sources of signals which are powerful enough to be detected and may, therefore, influence the measurements. These signals are considered “noise” for the purposes of train detection but, as expected, do not possess the characteristics of random noise (and never will). These are in fact signals that should and are being measured but “interfere” with the signal being searched for, which is a train passing.

It should also be clear that even though there is a relation between the channel number and its linear position alongside the track, the fiber route may not strictly follow the track and so the total fiber length can exceed the total track length. Many channels are located inside stations (go in and out) or are in fact parts of cable slack (loops) at certain positions along the path. These channels still report some signals but should, in general, be discarded as they don’t directly contribute to the calculation of the train movement. Measurements were made at every pole in the path in order to map the channel number to each pole.

All algorithms and techniques used here were developed to be used in almost “real time”, i.e., the train tracking should be done using a predefined maximum latency after receiving the raw data and should be executable by using “regular” contemporary workstations.

4.4 Intra Channel Analysis

The intra channel analysis uses the raw data for each channel in an independent manner in order to detect the passage of a train. It is, in fact, concerned about distinguishing periods of time when the incoming signal is not noise, i.e., it reports the current status of each channel in relation to noise.

At every time step each channel will be represented by a binary value indicating the detection or not of some sort of signal which is different enough from white noise.

4.4.1 Channel Raw Data

We have chosen to depict 3 channels which present different characteristics in order to demonstrate some of the problems and difficulties that were encountered and also to motivate the search for different measures that could be used in a simple and consistent manner and that do not rely on the magnitude of the signal.

Each channel will present varying amplitudes whose values are not know a-priori and neither can be trusted to stay within a measured range for all times. This is expected due to differences in the signal source (train) and also on the unpredictable and varying conditions of the fiber optic cable and its environment, e.g. soaked soil due to rain in contrast with dry soil, temperature variations, etc. will influence the attenuation of the signal until it finally reaches the measuring point.

Due to the scaling used in the following figures, the edges look sharp but, in fact, trains can be detected from quite far away, i.e., for each channel, the low frequencies are “heard” much before a train arrives and long after they are gone, albeit with increasing and decreasing average energy, respectively. This means that the observed signal envelope tends to gradually increase for the front end and decrease for the rear end on average, which means that we cannot use instantaneous values but we must average the values over a period of time and, hence, the “almost” real time nature of the whole system.

Figure 4-2 shows the raw data for channel 190. Just by glancing at this figure, it can be easily seen that there are 4 “short” intervals that stand out (where the signal oscillates with a much higher amplitude). These are, in fact, representative of trains moving through this channel. Of special interest is the magnitude (absolute value of the amplitude) the signal attains at these intervals, which is between 500.000 and 1.500.000 approximately (no units are reported).

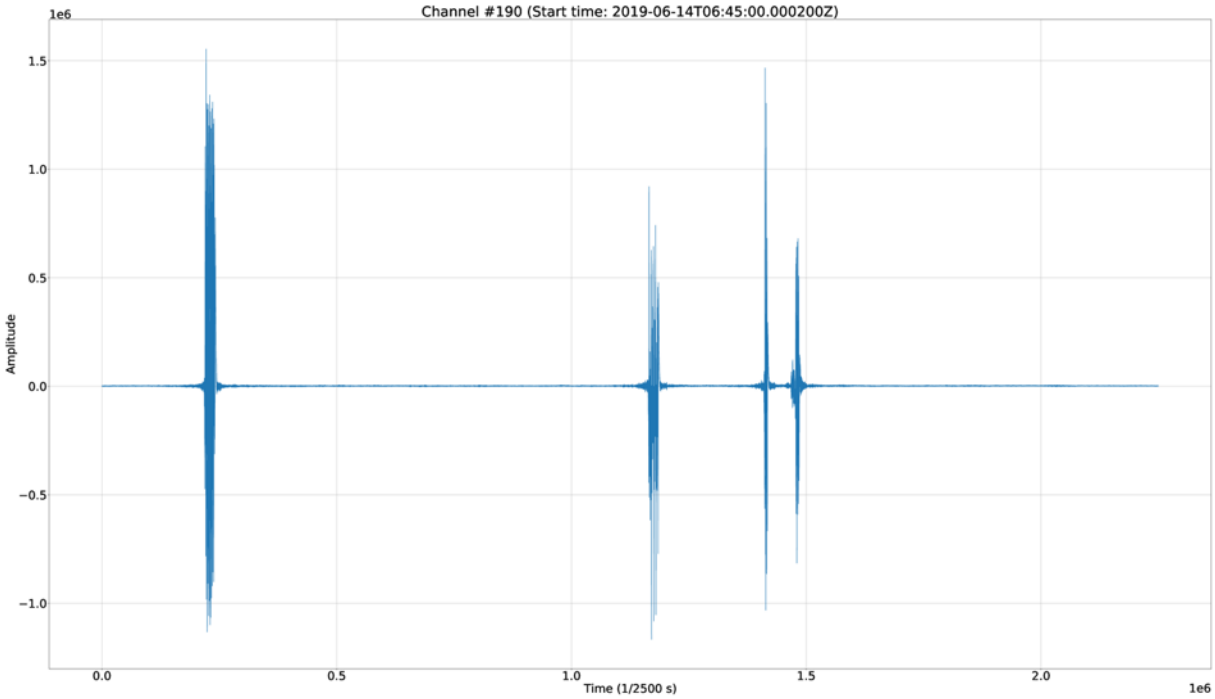

Figure 4-2: Channel 190 raw data

In contrast, Figure 4-3 shows the raw data for channel 820. Once again, we can clearly distinguish 4 “short” intervals where the amplitude is greater than the rest but, this time, not by such high factors as channel 190. In fact, the range goes from 10.000 to 30.000 approximately. Just like channel 190, this channel also presents a 3-fold increase from the low to high values but their absolute values are orders of magnitude smaller than the ones from channel 190.

Further investigation shows that this channel is in the Kissen train station and most probably the signal gets highly attenuated until it reaches the fiber optic cable. Figure 4-4 shows an aerial view of this section of the track.

It should be pointed out that the reported signal amplitude is dependent on the strength of the vibration at the measuring point which is a function, among other things, of the power of the source signal. This means that different amplitudes are expected depending on the length, weight, and speed of each train.

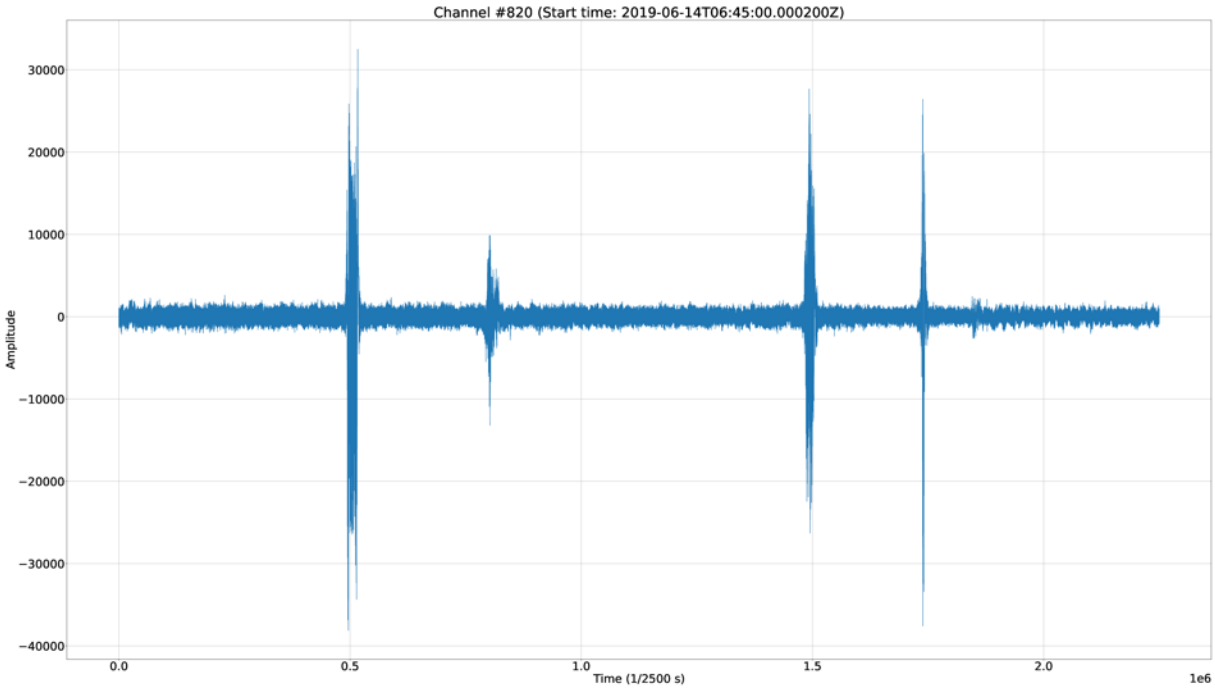

Figure 4-3: Channel 820 raw data

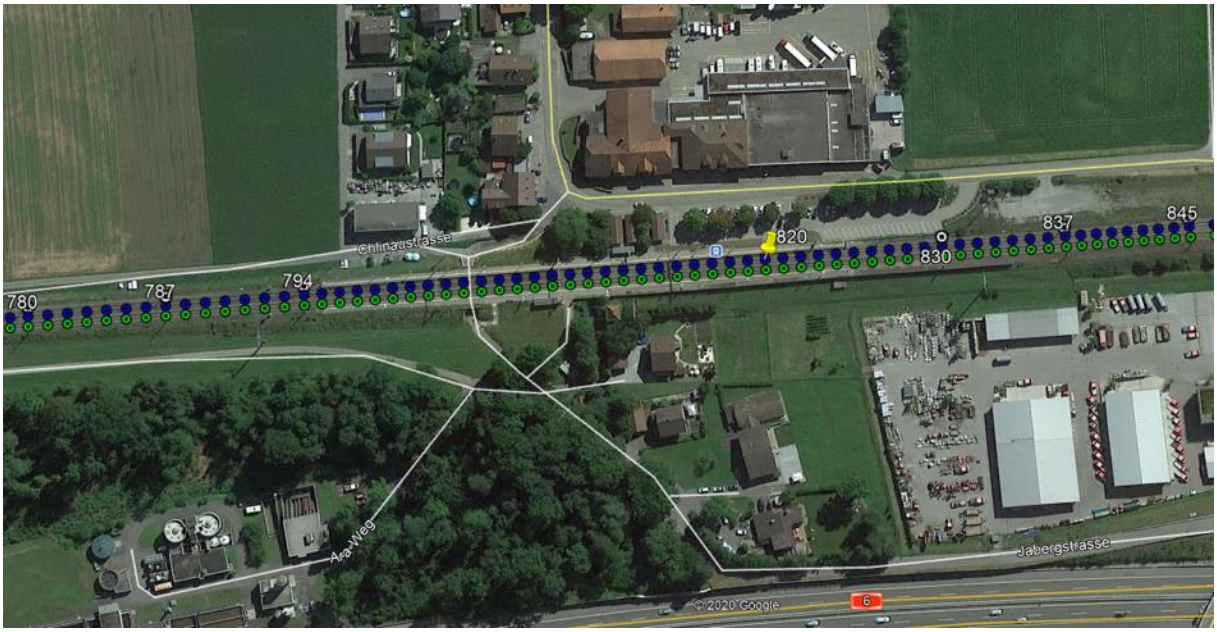

Figure 4-4: Kiesen station and respective channels

Finally, Figure 4-5 shows the raw data for channel 1050. The data for this channel looks a little different from the other 2, even though there are still 4 clear “short” intervals where the amplitude is disproportionately higher than the rest of the signal which are, in fact, due to the passage of trains. For this channel, the range goes from 100.000 to 300.000, approximately, which is an order of magnitude higher than channel 820 but half an order of magnitude lower than channel 190. Once again, the amplitude levels show a 3-fold increase, approximately.

This time, however, the “silence” intervals are not as “smooth” as the previous 2 channels and contain many “small” peaks which warranted further investigation.

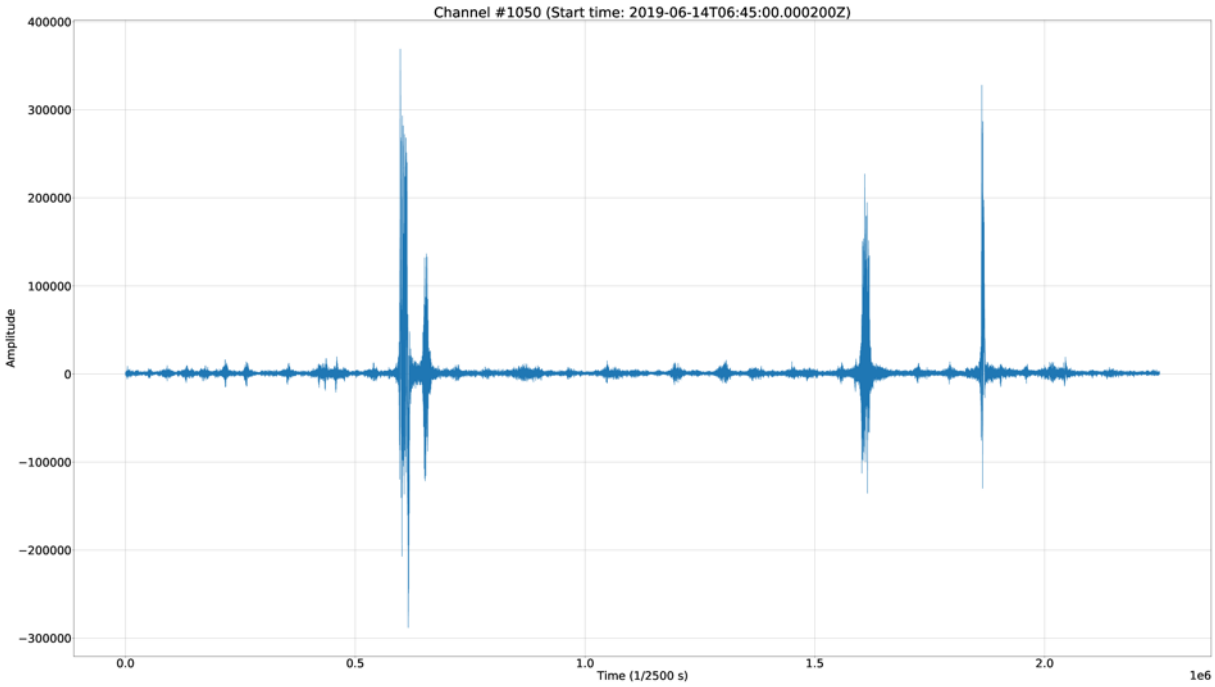

Figure 4-5: Channel 1050 raw data

These are, in fact, the signals produced by cars passing by on a highway that runs parallel to the tracks between channels 1030 and 1072 as can be seen on Figure 4-6.

Figure 4-6: Highway close to the train tracks

Even though these signals have a lower amplitude than the train signals, they are not “noise” under any classical definition and do present problems for train detection based on signal energy as they require adaptive algorithms with constant monitoring of energy levels in order to reliably filter them out.

4.4.2 Signal Energy and Power

Given a sequence of consecutive samples of length N from the original infinite sequence, the energy E_k of a compactly supported discrete signal $\{x_n\}$ in an interval of length N that goes from sample k up to $k + (N - 1)$ is defined as the sum of the squares of its values in this interval, i.e.,

$$E_k = \sum_{n=k}^{k+(N-1)} x_n^2$$

This sequence of samples can be viewed as the multiplication of the original signal by a rectangular window of length N starting at sample k. Assuming the independent variable n denotes time, we can define the discrete power P_k of this signal as simply the mean value of its energy in this interval, which will have the dimensionality of energy over time, i.e.,

$$P_k = \frac{E_k}{N} = \frac{1}{N} \sum_{n=k}^{k+(N-1)} x_n^2$$

The power as defined above make the measure more robust in relation to the window length, allowing for comparable levels for different window lengths.

Both the energy E_k and power P_k of a signal are, therefore, positive values and are usually converted to a logarithmic scale known as decibel (**db**) which is defined as

$$\boldsymbol{S}_{db} = 10 \log_{10}(S)$$

to better cope with the large dynamic range (many orders of magnitude) arising from these simple computations.

The signal power can be used as a first measure to analyse the signal and can, in fact, be used in order to generate the necessary events for train detection. It is probably the simplest and fastest way to detect the passage of trains as long as the silence power levels for each channel are below a certain threshold, which can be derived from sample data for each channel in an initial run.

4.4.3 Section and Step Length

The window length N plays an important role in the analysis of the signal. For accuracy, it is desirable to use the shortest window length possible that still reflects the low frequency components which are important.

The step size k, which is the number of samples to move the window, reflects the frequency with which the results are reported and usually allows for overlapping windows.

This means that the power calculations for the next windows will reuse many samples from the previous windows, effectively working as a lowpass filter and blurring the results by a certain amount.

Figure 4-7 shows the resulting plot when $N = 2500$ and $k = 250s$ where $s = 0,1,2, \ldots$, that correspond to 1 second long windows at every 100 ms. In this case, there are 8991 steps, 1 for the first second and 10 for each of the remaining 899 seconds of the 900 second interval.

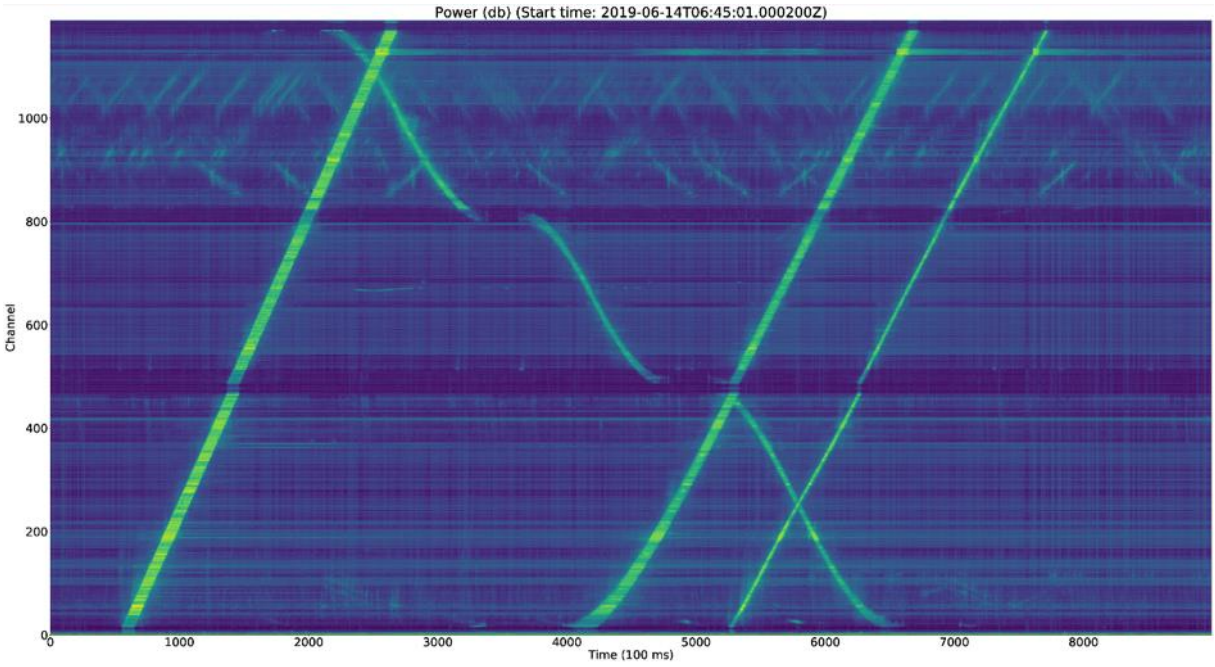

Figure 4-7: Raw data power using a 2500 sample (1 sec) window and a 250 sample (0.1 sec) step

The bright traces are clearly recognizable and stand out from the mostly dark but noisy background. It can be clearly seen that there were 3 trains going from Münsingen (channel 0) in direction to Thun (channel 1189) which move at almost constant speed (slope) and another moving in the opposite direction which makes 2 stops (accelerates and decelerates).

As expected, the trains “disappear” when they are not moving, as can be seen when the train moving from Thun towards Münsingen stops at each station.

Many other features also stand out from the noisy dark background:

- Many smaller inclined lines with both positive and negative slopes are seen on the higher channels (where the highway is close to the tracks)
- Very low energy levels (very dark) at the Kissen station (around channel 820)
- Very low energy and signal “jump” at Wichtrach station (around channel 480) which indicates fiber slack (loop).
- Some sections with a “rectangular” shape which indicate that the channels are “tied” together, i.e., vibrate in unison (short bridge length and a few other places).
- Many vertical lines in the background

Using a different window length of 500 samples (200 ms) and the same step size of 250 samples (100 ms), the resulting power plot is shown in Figure 4-8.

Apparently, the differences at this scale are not very significant at first glance but we can expect the bright traces to be narrower for the shorter window length.

It should be noted that using 500 sample windows will result in 8999 steps for the 900 second (15 minute) interval as only the first step will be skipped. In order to be able to compare the results we have skipped the first 8 steps for this window size so that both windows are aligned in time and start after 2500 samples have been gathered.

Both plots can be interpreted as the power calculated with a window which ends at the current sample, i.e., using the previous N samples up to the current one and both start at the same time, which is 1 second after the start of the sample 15-minute interval used for the current report.

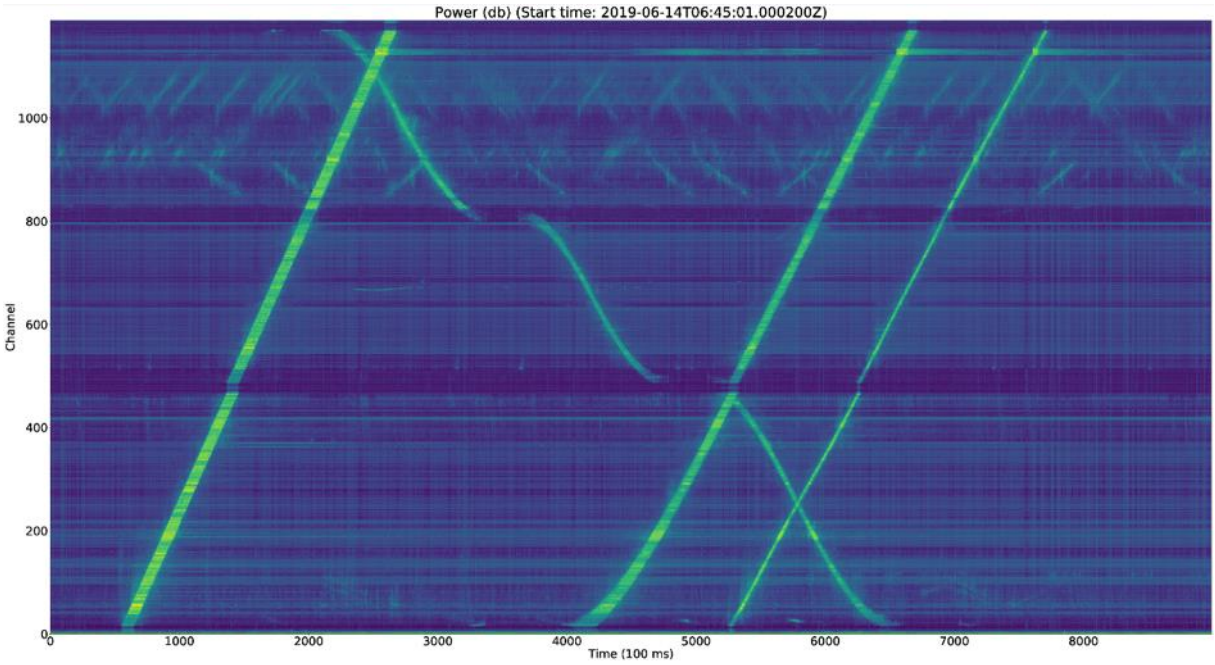

Figure 4-8: Raw data power using a 500 sample (0.2 sec) window and a 250 sample (0.1 sec) step

Figure 4-9 shows the power difference between the 2500 and 500 sample windows, which makes it apparent that the edges present larger differences and confirms the hypothesis of shorter trace widths using a narrower window. Also, narrow windows will produce sharper edges but will also present more noise in both the background and signal areas.

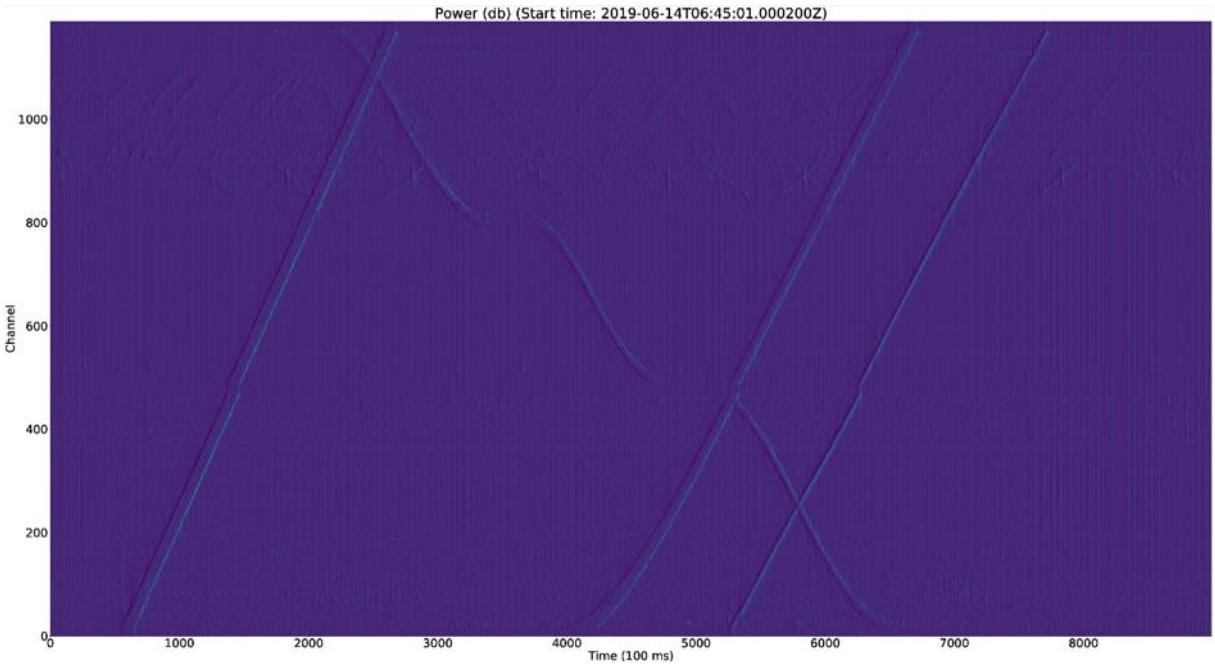

Figure 4-9: Difference between the raw 2500 sample (1 sec) and the 500 sample (0.2 sec) window

In these pictures, low values are represented with a darker shade while high values are represented by a brighter shade. It can be seen that, horizontally, there is a darker line that always precedes a bright one in time. This means that a longer window will present lower values for the train front and higher values for the train rear.

4.4.4 Highpass (DC notch) Filter

Some channels present a slowly varying fixed offset (DC level) that should be removed as this greatly affects the power calculations and induce large errors at the thresholding stage. This means that some sort of filtering needs to be done on the original signal in order to remove these very low frequency components before calculating the power of the signal.

A simple DC notch filter should be enough to remove DC and very low frequencies and its design requires the specification of a single parameter. One possible design parameter is the specification of the filter's cutoff (half power) frequency which leads to the desired filter rate which can then be used for its implementation.

As an example, using a rate of 0.995 (adaptation constant of 200 samples) would create a DC notch filter with a cutoff frequency of around 2Hz (1.99442Hz) using a 2500Hz sampling rate. Also, the removal of some very low frequencies should help with the removal of some periodic noise that could be responsible for some of the vertical artefacts seen on the previous power plots.

Figure 4-10 shows the resulting plot using the 2500 sample window on the DC notch filtered signal. Comparing it with Figure 4-7, it is clear that the background has lower power levels (is darker), indicating that there was a significant amount of power that was derived from these very low frequencies which was removed using the DC notch filter.

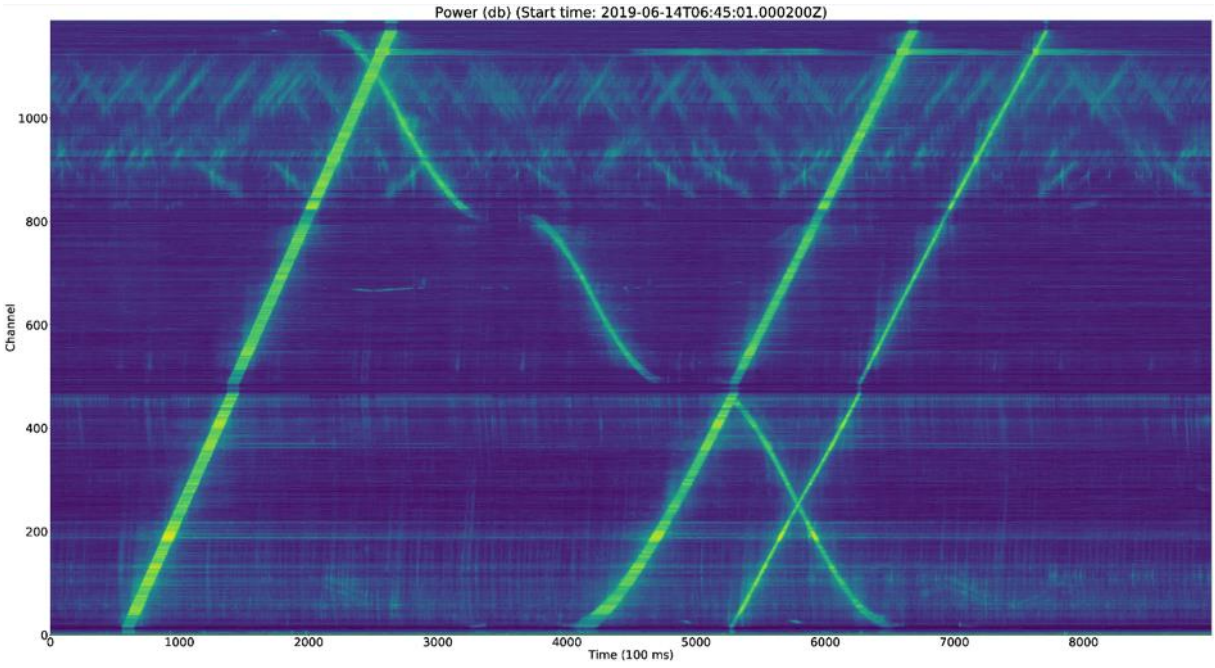

Figure 4-10: DC notch filtered data power using a 2500 sample (1 sec) window and a 250 sample (0.1 sec) step

Figure 4-11 shows the resulting power plot using the DC notch filtered signal with 500 sample long windows. The same remarks can be applied here, when comparing it with Figure 4-8. Once again, at first glance it is not easy to distinguish the differences between these 2 plots but a careful inspection shows the blurring caused by the longer window length.

For completeness, Figure 4-12 shows the difference between the power plots for the 2500 and 500 sample long windows when using the DC notch filtered signal. Careful comparison with Figure 4-9 reveals a much smoother background, showing that most vertical artefacts due to low frequency components have been removed from both window lengths, otherwise they would have to appear as a dark and bright pair of lines in the difference plot.

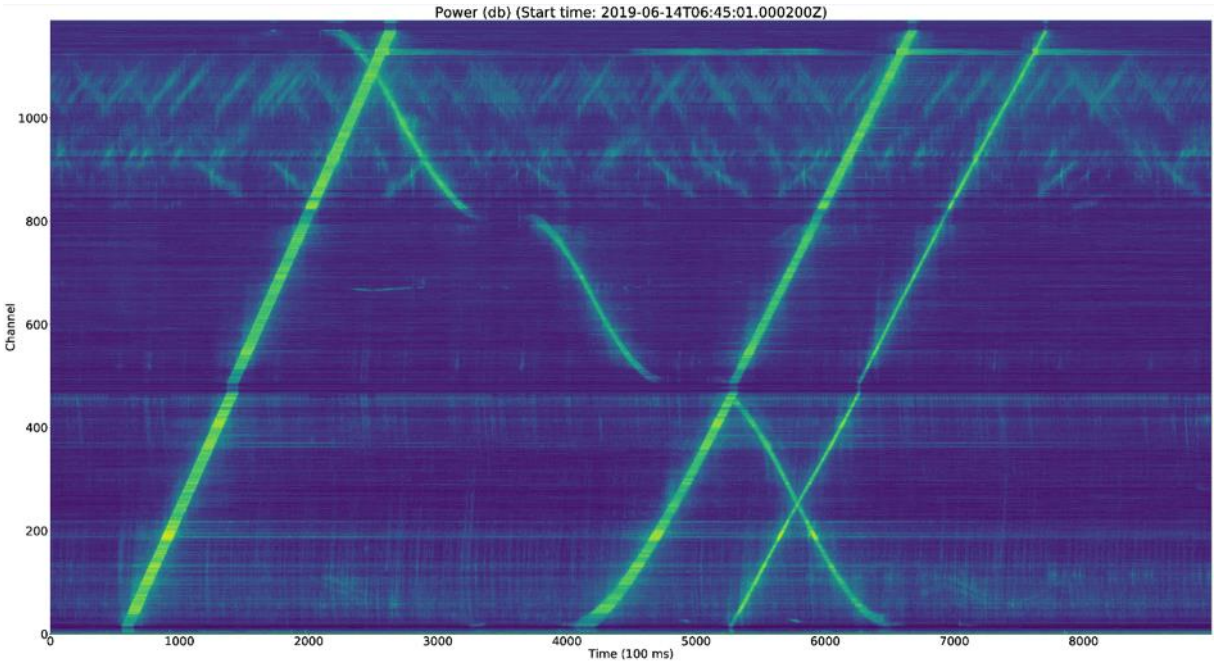

Figure 4-11: DC notch filtered data power using a 500 sample (0.2 sec) window and a 250 sample (0.1 sec) step

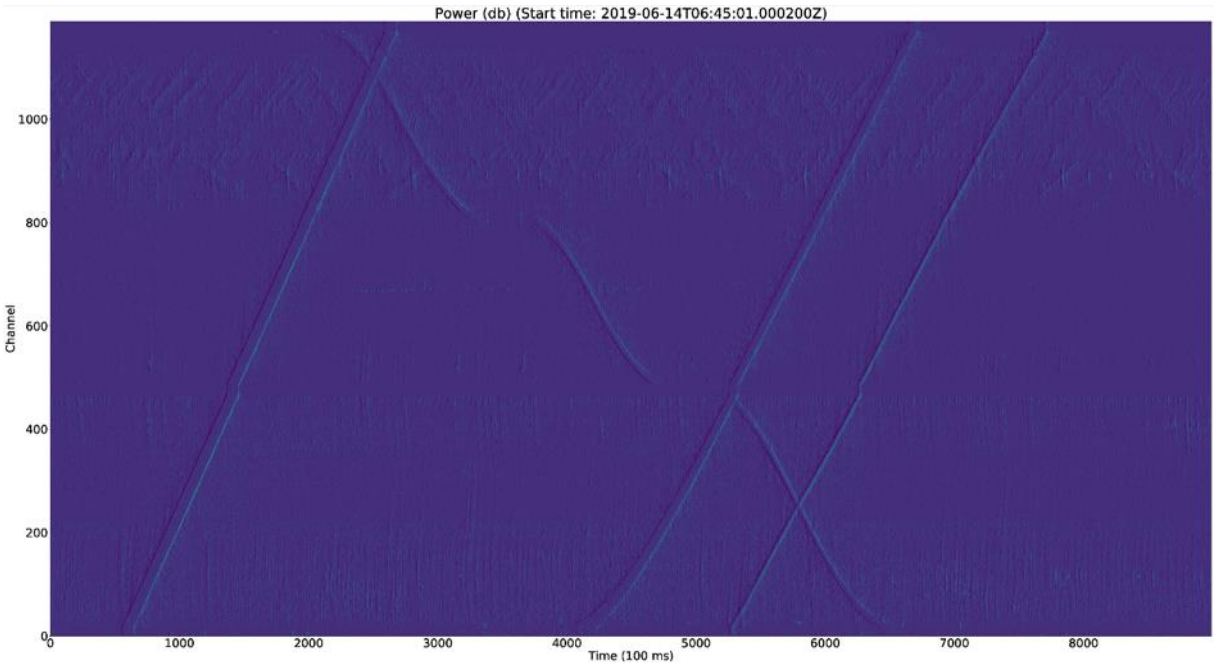

Figure 4-12: Difference between the DC notch filtered 2500 sample (1 sec) and the 500 sample (0.2 sec) window

It should be noted, however, that the lower channels still exhibit quite a large amount of low frequency power (bright background) which suggests that a stronger high pass filter should be applied. However, the amount of filtering should be just enough in order to make the background behave like white noise, if possible, as it also eliminates important signal information.

4.4.5 Spectrum Analysis

Better tools for dealing with both the noise and the data are available by transforming the signal into the frequency domain.

By increasing the computational power required to process the signal but still keeping it well within the bounds of current technology to do this in real time, we can calculate the power spectrum density (PSD) for each channel which, in turn, can be manipulated with the objective of eliminating noise and undesired signals as well as allowing the use of other metrics for thresholding which are invariant to scale (amplitude independent).

The PSD will decompose the signal power into its constituent frequencies which allows us to basically try to remove our dependency on the magnitude of the spectrum (amplitude) and concentrate on its distribution of energy among its frequencies (profile) expecting that train signals present an entirely different profile from the one from noise.

The basic tool for frequency analysis is the Discrete Fourier Transform (DFT) which possesses a fast algorithm for its computation known as the Fast Fourier Transform (FFT). Some details about the DFT which are worth mentioning are:

1. Both the time and frequency sequences are periodic with period given by the interval length: an implicit assumption given by the nature of the DFT is that the input sequence is periodic and so is the transformed sequence;
2. Time localisation is lost: the values of the transform give the frequency components for the whole periodic signal and altering any frequency coefficient will not result in a coherent signal in time (aliasing in time) - if the intention is to go back to the time domain (perform an inverse transform) then altering the frequency coefficients will most probably not result in a meaningful time signal;
3. Energy conservation: the energy or power of the signal is preserved which means that calculating the power using the original signal samples or the transform coefficients should yield the exact same results (Parseval's theorem).

4.4.6 Signal Filtering

As expected, the use of a DC notch filter helps with removing unwanted noise. However, as we have no intention to ever reconstruct a time signal but are mostly interested in thresholding a measure which indicates that a signal is present, we can filter unwanted frequencies in a much more efficient and easier manner by just eliminating the DFT bins which correspond to the unwanted frequencies (filtering in the frequency domain).

In fact, our main goal is to be able to properly characterize the background noise, i.e., filter the signal in such a way that the resultant frequency profile resembles white noise in the time intervals when there is no vibration (no significant object moving).

Also, we can expect to eliminate weaker signals by applying stronger filtering, i.e., eliminating more frequency bins, assuming that meaningful higher harmonics will only be present in a significant amount for stronger signals, which generate stronger harmonics.

4.4.7 Power Spectral Density Estimation

The signal power can be better estimated and broken down into its constituent frequencies by using the magnitude squared of its DFT coefficients. As energy is preserved by the DFT, the final power using the transformed coefficients should be exactly the same as the original power calculated using the original coefficients.

Having the contribution of each frequency, however, allows us to filter out the frequencies that are too noisy to be useful and also concentrate on the highest frequencies possible, as these will provide sharper edges for thresholding.

Also, the availability of the PSD allows us to move away from the simple power calculations and into different measures which are based on the normalized PSD, so that the magnitude is of little relevance for the calculation of the measure.

In fact, we may regard the problem of train detection using FOS as an audio activity detection and apply some of the methods used for speech detection in this context.

4.4.8 Periodogram

The periodogram is the building block of many PSD estimation techniques and consists of the squared magnitude of the DFT coefficients, i.e., the periodogram is nothing more than the magnitude squared coefficients of the Short Time Fourier Transform (STFT).

The main problem with a single periodogram for PSD estimation is that the variance at a given frequency does not decrease when the number of samples N increase.

In fact, even if the estimate did get better for longer N (which is not the case), we would still only be interested in the shortest value of N possible to reduce the blurring caused by longer signals, as we are using the whole block of N samples as the time interval for the purposes of edge detection and thresholding.

There are other methods which use many periodograms and average their results that yield better estimates. These methods usually achieve lower estimate errors by reducing the resulting frequency resolution.

4.4.9 Welch Power Spectral Density Estimation

The method of averaged periodograms, also known as the Welch method, trades PSD noise by frequency resolution. It reduces noise in the estimated PSD in exchange for reduced frequency resolution. In general, the variance of each frequency bin is reduced by the number of periodograms averaged.

It works by dividing the signal into overlapping segments, i.e., the original signal is split up into L data segments of length M, overlapping by D samples. This means that $(N - M) = (L - 1)(M - D)$, where N is the signal length.

Using the raw signal (without any filtering) with the following parameters: $N = 500$, $M = 250$, and $D = 125$ results in $L = 3$ segments of 250 samples in length which overlap by 50% (125 samples) generating 3 periodograms which were then averaged resulting in the final PSD estimate for this 500 sample signal. These values were used for the examples presented, but different values using a higher number of segments would yield better estimates. Also, using a segment length which is a power of 2 might speed up the calculations by quite a significant margin for most FFT implementations.

As the original signal is being simply truncated, this is the same as multiplying the original infinite signal by a rectangular window of length 500.

The resulting PSD (in db) for channels 190, 820, and 1050 are plotted in Figure 4-13, Figure 4-14, and Figure 4-15, respectively. The train passages consist of brighter vertical strips which can be identified with relative ease for the first figure but with more difficulty for the other 2.

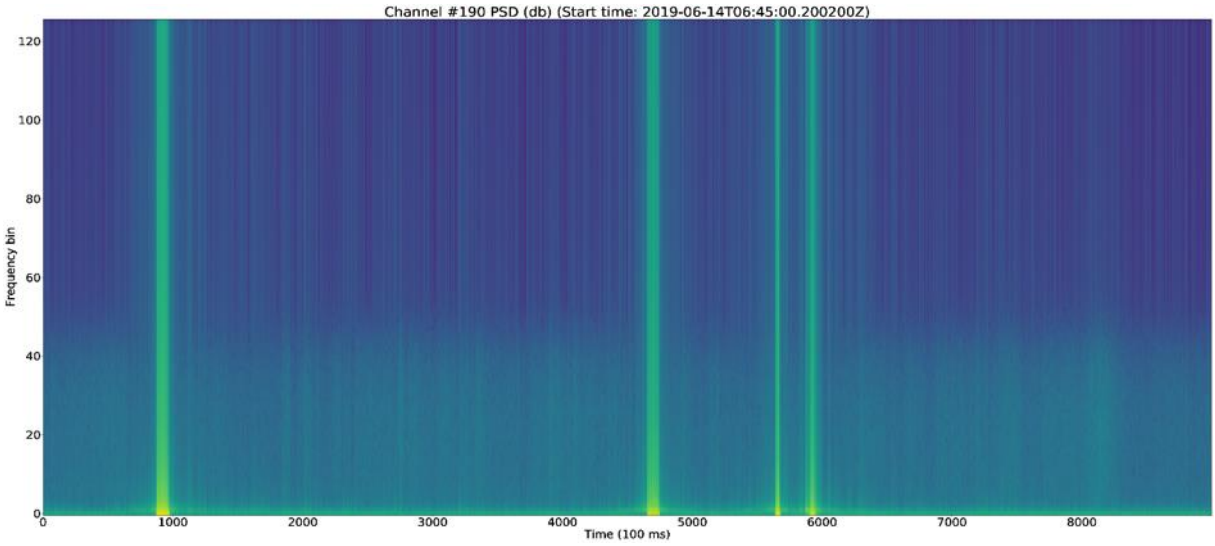

Figure 4-13: Channel 190 PSD (db) 500 (3 x 250 + 125) x 250 (rectangular)

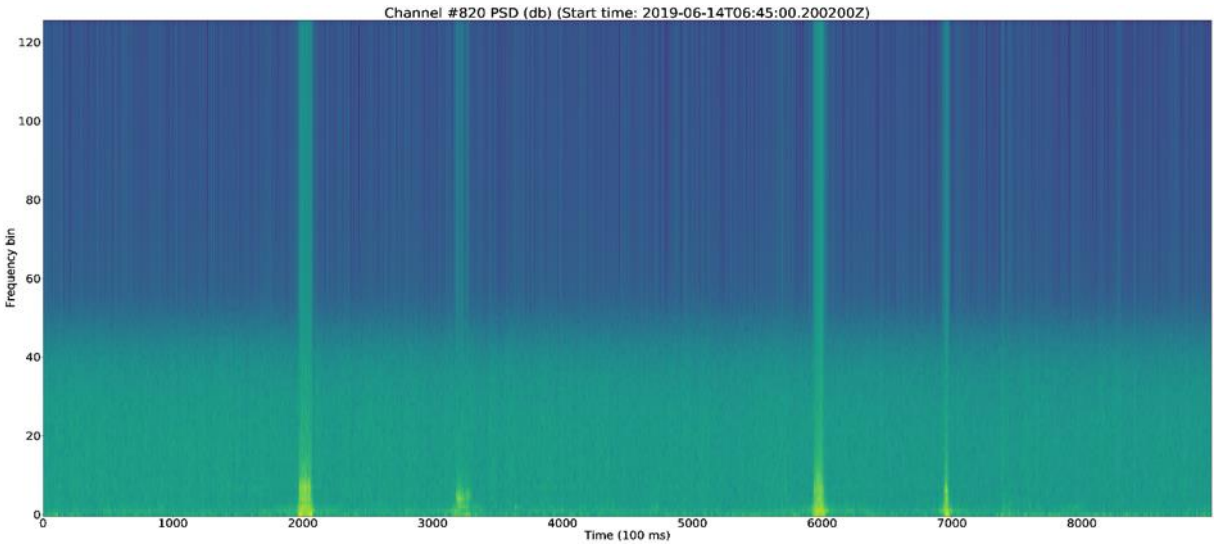

Figure 4-14: Channel 820 PSD (db) 500 (3 x 250 + 125) x 250 (rectangular)

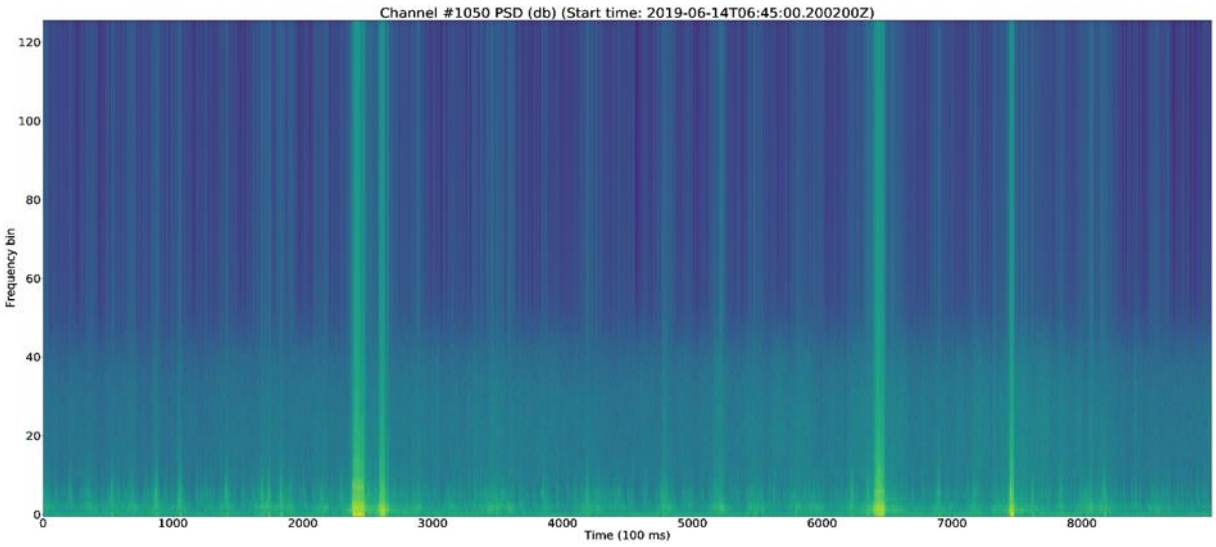

Figure 4-15: Channel 1050 PSD (db) 500 (3 x 250 + 125) x 250 (rectangular)

Due to the use of a rectangular window, spectral leakage occurs and all plots present many thin vertical lines that should not really exist.

4.4.10 Spectral Leakage

Spectral leakage is a broad term which refers to the power of some frequencies “leaking” into neighbouring frequencies.

This could be caused by sampling itself, in which case it is also referred as aliasing, but more frequently it is due to the multiplication of the original time series by a function which is zero outside the desired domain (window function). Multiplication in time results in convolution in the frequency domain so that the net result is that the signal DFT gets convolved with the window function DFT.

Spectral leakage cannot be entirely eliminated but it can be reduced by using appropriate windowing techniques before calculating the DFT. In fact, window functions allow for the distribution the frequency leakage spectrally in different ways, according to each need.

4.4.11 Windowing

A Window function, also known as an apodization function or as a tapering function, is a function used to smoothly bring a sampled signal down to zero at the edges of the sampled region. This suppresses leakage side-lobes which would otherwise be produced upon performing a DFT, but the suppression is at the expense of widening the lines, resulting in a decrease in the frequency resolution.

The rectangular window, which has been implicitly used so far, does allow for high energy leakage into frequencies which are quite distant from the real ones and does not produce good results.

There exists a huge number of windows which can be used depending on the application and most of them present a large improvement in relation to the rectangular window regarding frequency leakage. A good general-purpose window is the Hann window which is defined as:

$$w(n) = \frac{1}{2}\left(1 - \cos\left(\frac{2\pi n}{N}\right)\right)$$

The Hann window was used for the examples presented here but there may be other windows which produce slightly better results, e.g., the Blackman window. The difference, however, is somewhat subtle and the choice of window is largely open to debate and results using different windows should be compared.

Figure 4-16, Figure 4-17, and Figure 4-18 show the resulting plots of the PSD for channels 190, 820, and 1050, respectively, using a Hann window.

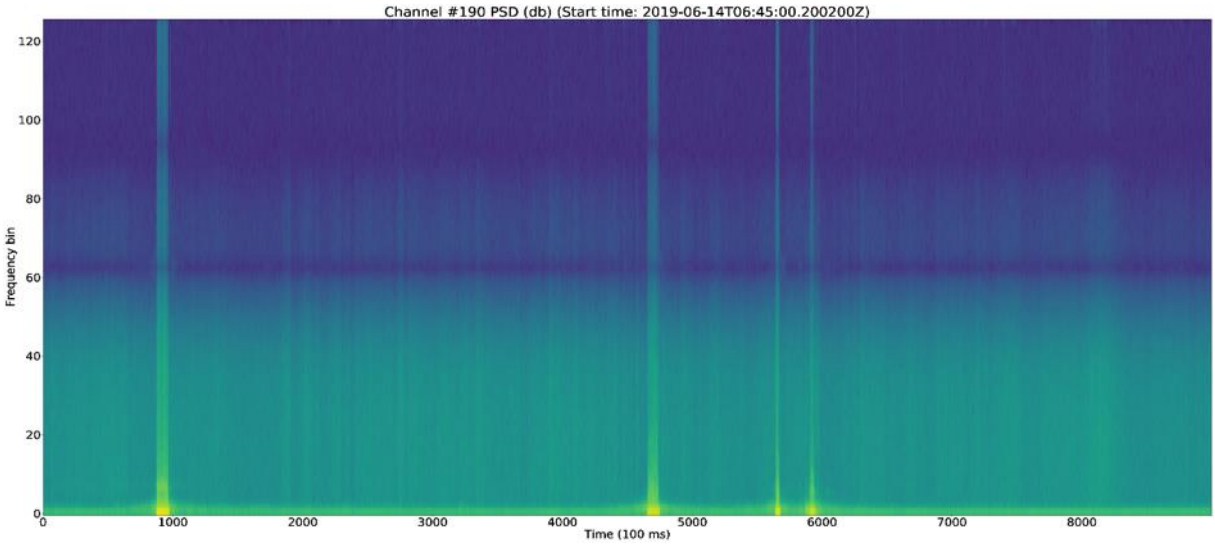

Figure 4-16: Channel 190 PSD (db) 500 (3 x 250 + 125) x 250 (hann)

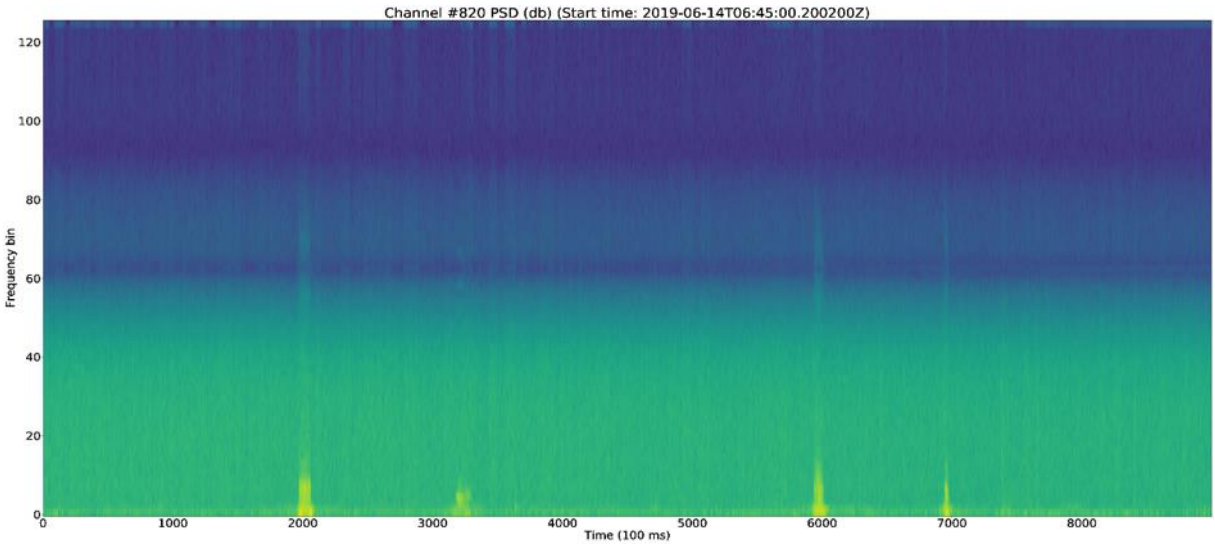

Figure 4-17: Channel 820 PSD (db) 500 (3 x 250 + 125) x 250 (hann)

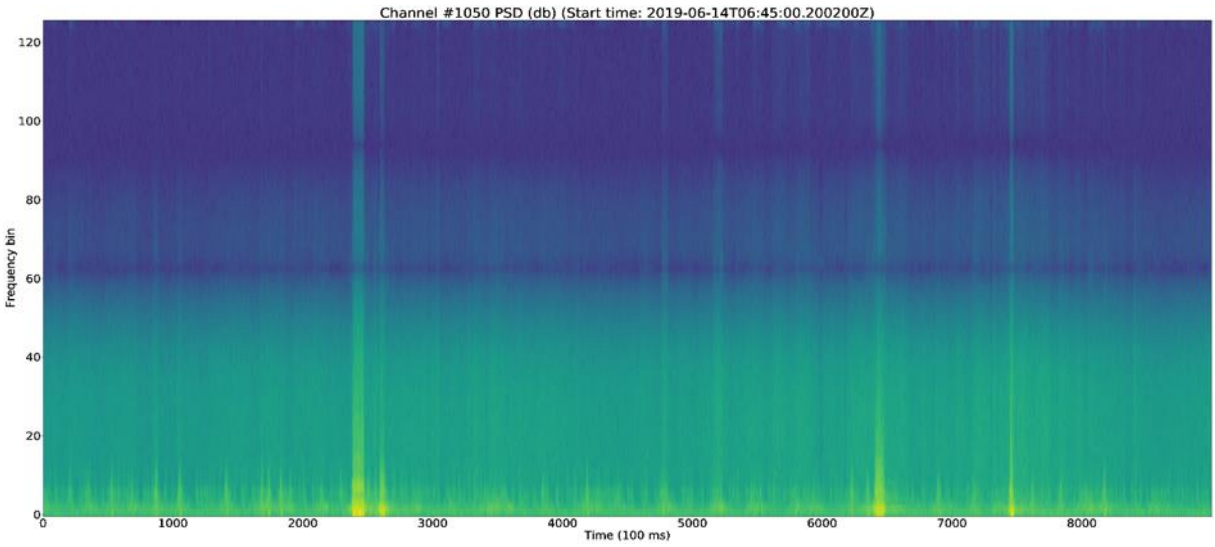

Figure 4-18: Channel 1050 PSD (db) 500 (3 x 250 + 125) x 250 (hann)

Comparing these with the ones using the rectangular window, there is a marked improvement in the leakage (most thin vertical lines have disappeared) but at the expense of a less distinct and more blurred signal.

Also, when using the Hann window, some frequency bins get highly attenuated and show up as dark horizontal lines in the above 3 plots. These frequencies are exactly 1/4, 3/8, … of the sampling frequency or 1/2, 3/4, … of the Nyquist frequency.

For completeness, the PSD for all channels using the Welch method with both a rectangular and a Hann window are shown in Figure 4-19 and Figure 4-20, respectively.

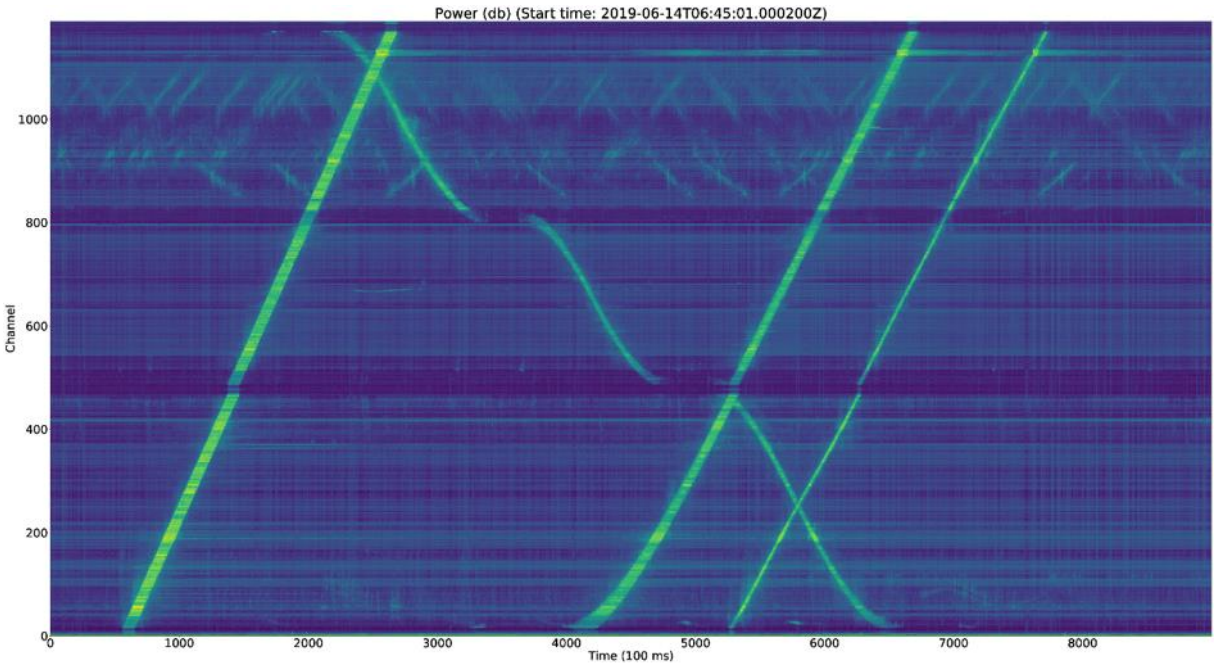

Figure 4-19: Welch PSD (db) estimate 500 (3 x 250 + 125) x 250 (rectangular)

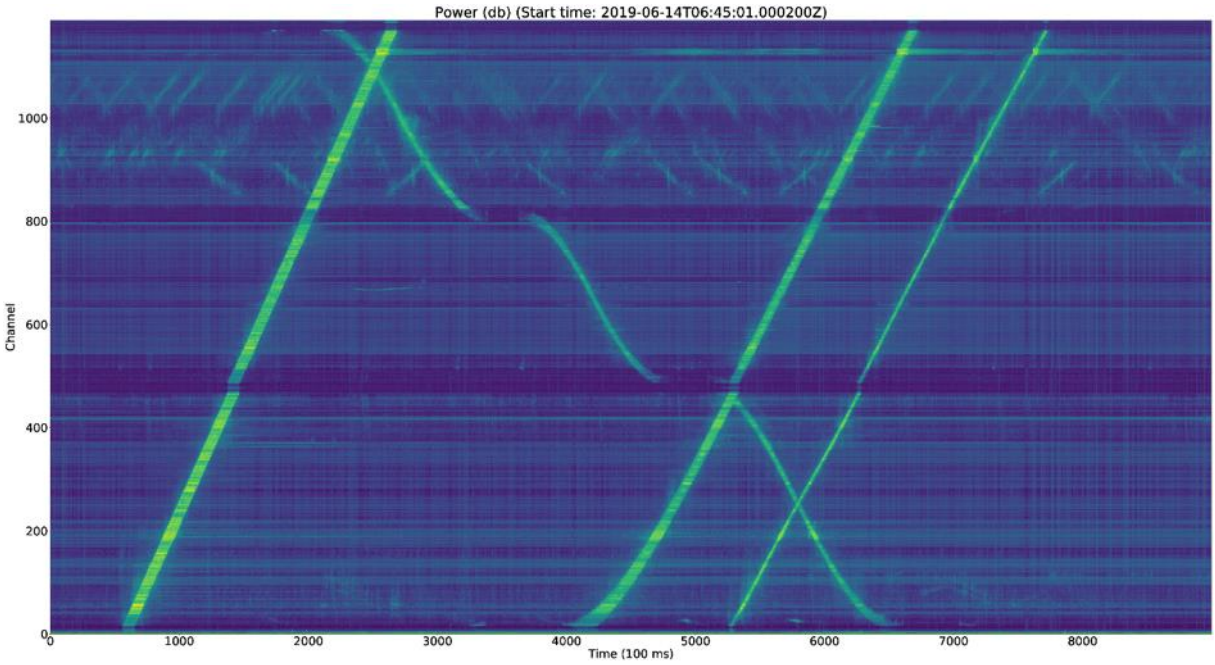

Figure 4-20: Welch PSD (db) estimate 500 (3 x 250 + 125) x 250 (hann)

These 2 plots should look alike, as they use the whole energy from all frequencies. In fact, these 2 plots should look pretty much the same as Figure 4-8, which is simply the power plot using the same parameters.

4.4.12 Frequency Domain Filtering

As discussed before, there is no intention of going back to the time domain so we can analyse the signal power in the absence of some frequency bins. In fact, we can eliminate most of the power contributed by the low frequency components by simply setting their DFT magnitude to 0 and computing the power of the resulting signal.

Also, as discussed with the manufacturer of the sensing equipment (OptaSense), there is not much information collected above a quarter of the sampling frequency (2500 / 4 = 625 Hz) which would be attenuated in any case by the use of the Hann window. Therefore, we have also chosen to eliminate all frequencies above a certain value.

The net result is a bandpass filter done in the frequency domain by eliminating some low and high frequency bins from the DFT.

The resulting PSD plots using only frequency bins 6 to 40 which is equivalent to a bandpass filter from 60 to 400 Hz are shown in Figure 4-21 and Figure 4-22 for both the rectangular and Hann windows, respectively.

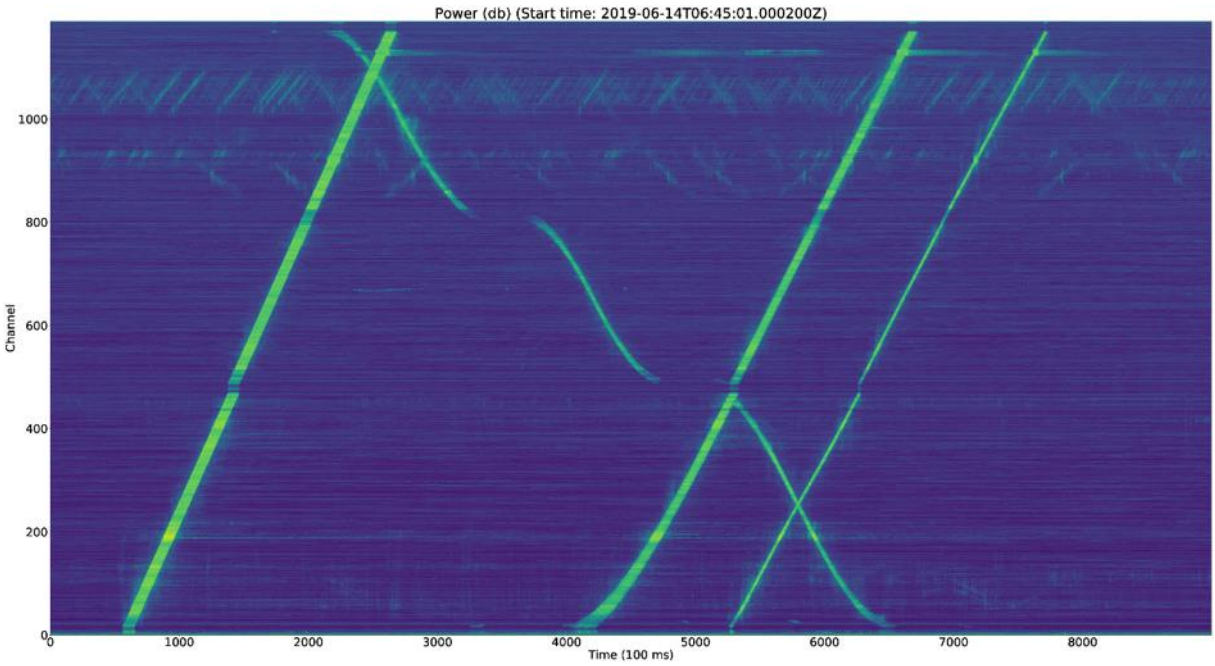

Figure 4-21: Welch PSD (db) estimate 500 (3 x 250 + 125) x 250 (rectangular) from 60 to 400 Hz

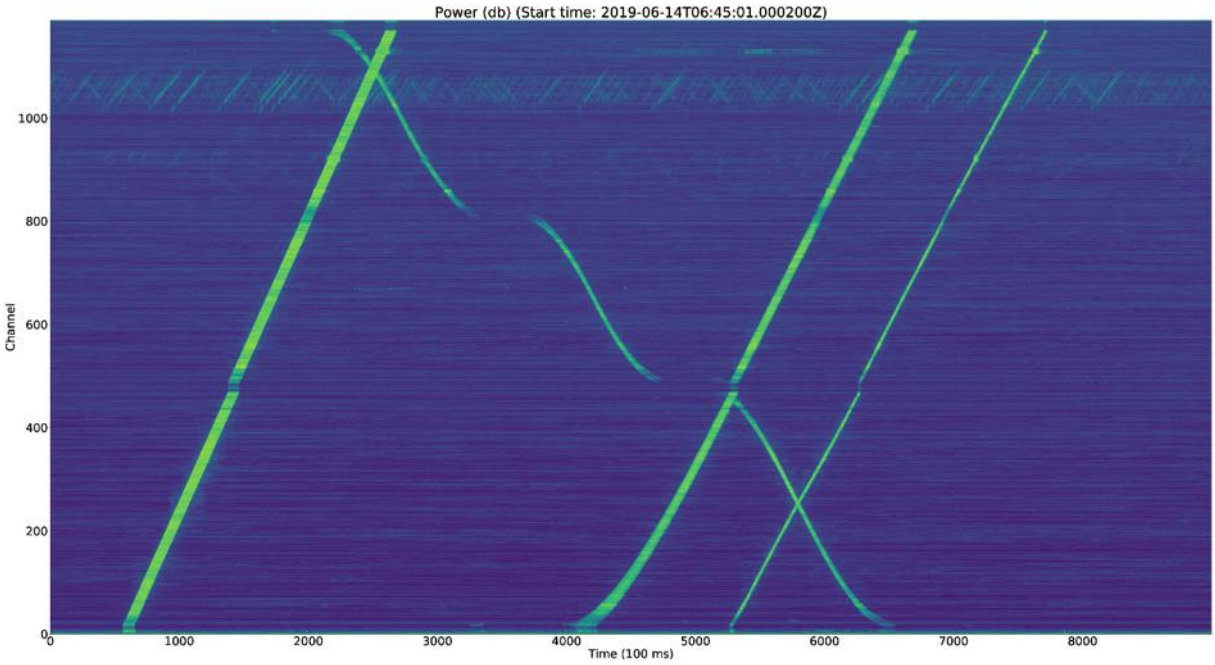

Figure 4-22: Welch PSD (db) estimate 500 (3 x 250 + 125) x 250 (hann) from 60 to 400 Hz

Most background noise is gone, which now looks more uniform. Also, there is a marked difference when using the Hann window when compared to a rectangular window, mostly seen as better-defined edges and less contribution from the signal coming from the highway on the higher channels (less leakage).

Increasing the filtering from bins 12 to 40 (120 to 400 Hz), the resulting PSD plots are shown in Figure 4-23 and Figure 4-24 for both the rectangular and Hann windows, respectively.

Once again comparison between the rectangular and Hann window plots reveals that the lower power signals coming from the adjacent road is almost entirely gone when using the Hann window.

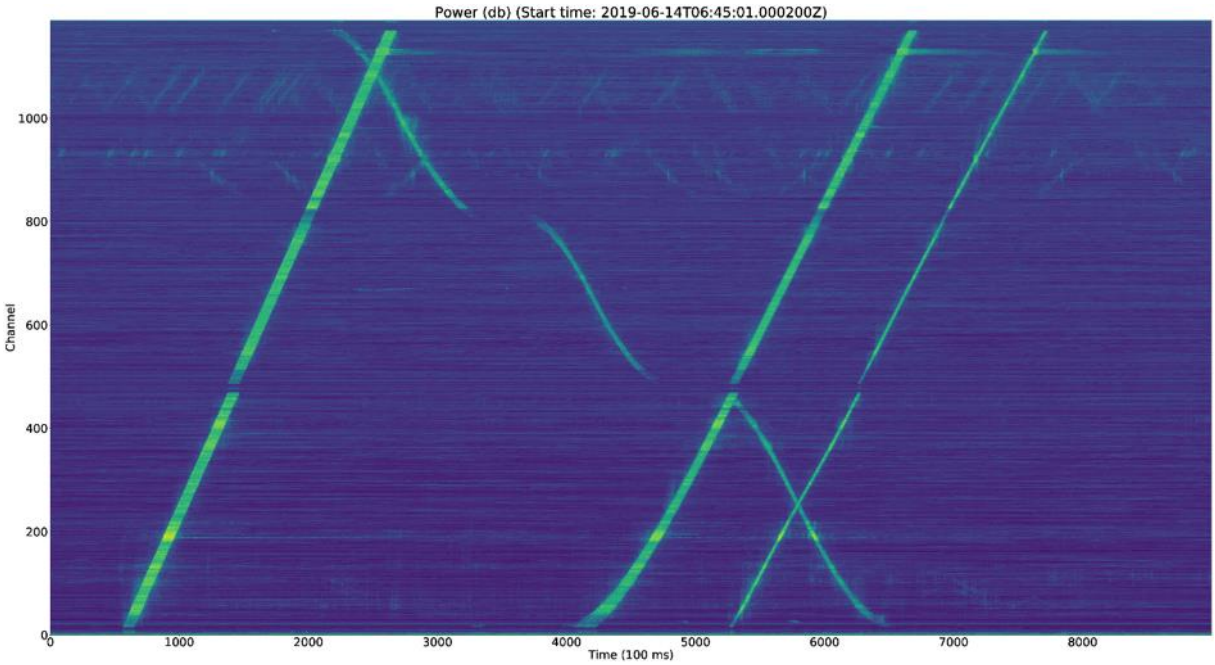

Figure 4-23: Welch PSD (db) estimate 500 (3 x 250 + 125) x 250 (rectangular) from 120 to 400 Hz

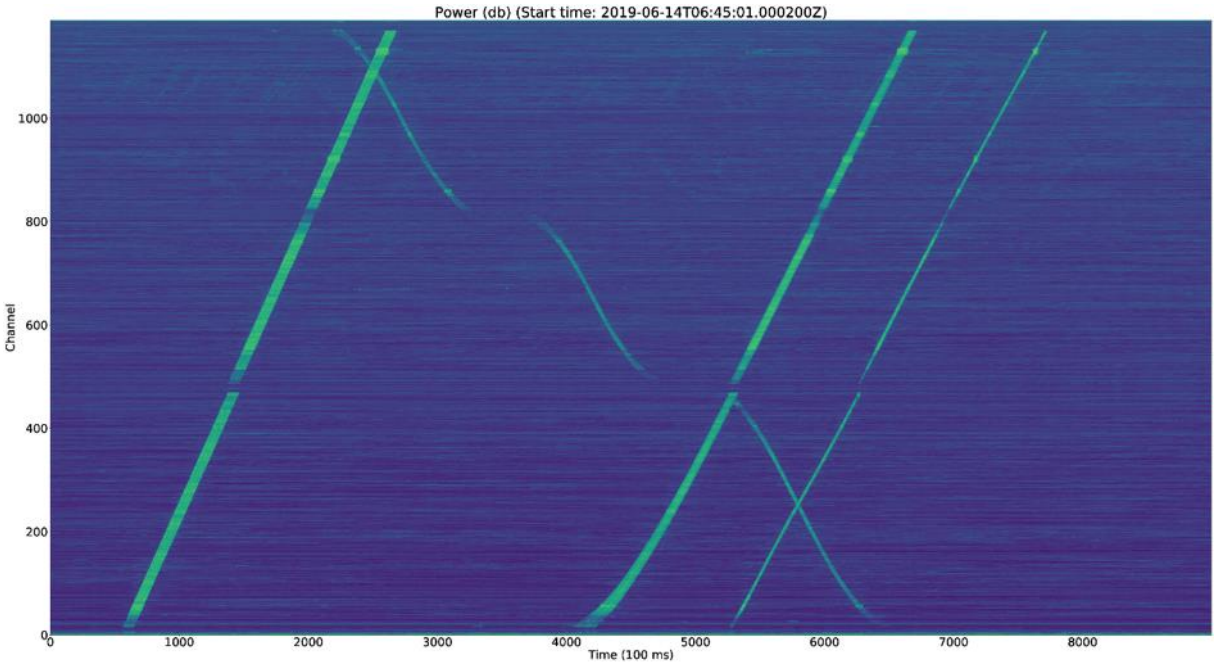

Figure 4-24: Welch PSD (db) estimate 500 (3 x 250 + 125) x 250 (hann) from 120 to 400 Hz

However, so much information has been eliminated that some channels are fading away, especially the ones which already had high attenuation, e.g., channel 820.

4.4.13 Spectral Flatness

As briefly discussed in the introduction, we are in search of a measure which is independent of the amplitude and can be used in an independent way, i.e., which has absolute values that can serve as measures for how close a signal is to noise.

Spectral flatness (SF), also known as tonality coefficient or Wiener entropy, is a measure to characterize an audio spectrum.

It is defined as the geometric mean divided by the arithmetic mean and produces a number between 0 and 1 (both the product and sum go from 0 to $N-1$):

$$\mathrm{SF} = \frac{\sqrt[N]{\prod x_n}}{\frac{\sum x_n}{N}}$$

The SF should be close to 1 for white noise and close to 0 for a signal composed of a single frequency. It should, theoretically, gradually go from 0 to 1 as more and more energy is distributed among the frequencies.

The SF, however, suffers from some serious numerical instabilities due to its numerator being the geometric mean, which will suffer greatly if a single coefficient is unusually low.

4.4.14 Entropy Spectral Flatness

A measure based on information theory which has the same characteristics of the spectral flatness but does not suffer from the same instabilities has been proposed in a paper from N. Madhu (Electronics Letters, 5th November 2009, Vol. 45 No. 23).

This measure, which we call the Entropy Spectral Flatness (ESF) is calculated by first defining the normalized sequence $\{\bar{x}_n\}$ defined as (where the sum is from 0 to $N-1$):

$$\bar{x}_n = \frac{x_n}{\sum x_n}$$

in that each $\bar{x}_n$ can be seen as a probability so that $\sum \bar{x}_n = 1$. The ESF can then be defined as (where the sum is from 0 to $N-1$):

$$\log_2(ESF + 1) = -\frac{\sum \bar{x}_n \log_2 \bar{x}_n}{\log_2 N}$$

Both the SF and the ESF rely on the background being composed of white noise and, therefore, should not be applied in situations in which this is not the case, which is shown in Figure 4-25 for completeness.

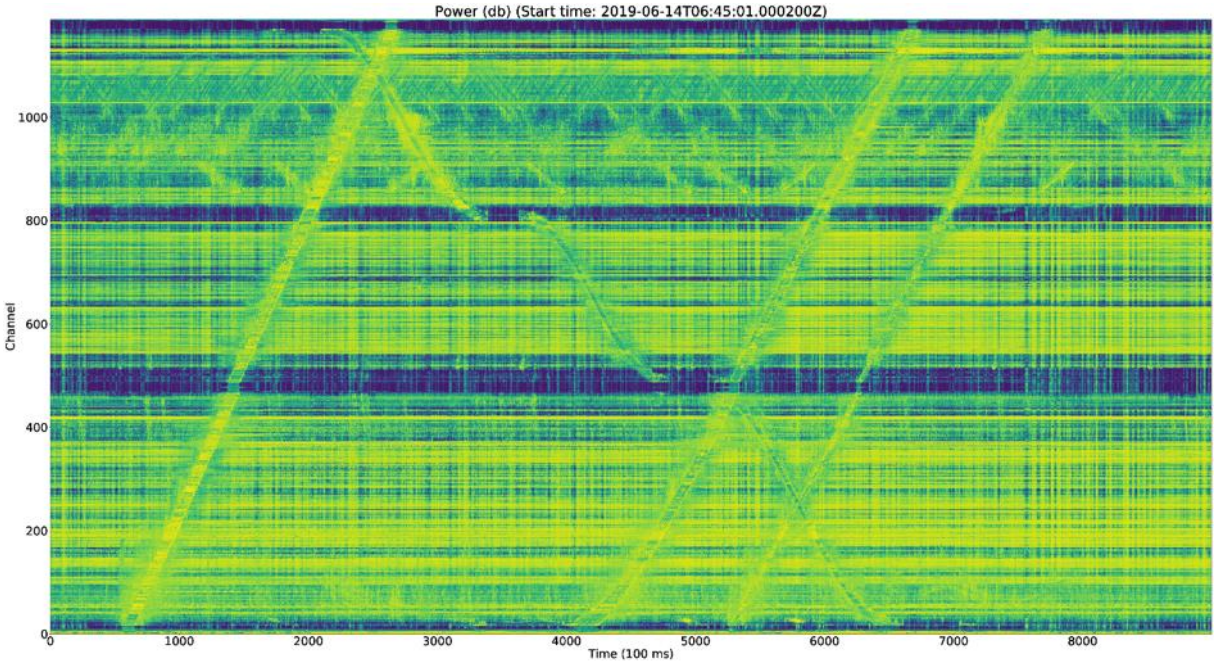

Figure 4-25: ESF Welch 500 (3 x 250 + 12) x 250 (hann)

Figure 4-25 makes it clear that the background noise is definitely not white noise. Filtering the signals, however, makes it clear that there are low frequency components across all channels that, once removed, make the background noise more similar to white noise.

As the ESF, just like the SF, is based on normalized values, large differences in the signal power are lost and only the relative contribution of each frequency bin is taken into account.

Figure 4-26 and Figure 4-27 show the ESF for the filtered signal from 60 Hz and from 120 Hz until 400 Hz, respectively. Both plots were made using the normal ESF range from 0 to 1, i.e., they were not converted to decibels.

For Figure 4-26, even though the background is now almost completely smooth, it is quite clear that the signals coming from the road around channel 1050 are also very noticeable. In fact, they have values which are on par as the ones from the train. Also, there is quite a large amount of noise in some channels as we get close to the train which means that the ESF is detecting them as they approach and leave this channel.

More filtering (from 120 Hz to 400 Hz), as shown in Figure 4-27, helps and makes most of the signal coming from the road to disappear but also, as expected, weaken the signal from the trains, especially for those channels that already had a low dynamic range to start with. It also eliminates most of the early detection and late dismissal caused by high powered low frequencies which can be heard from far away, as expected.

This suggests that the amount of filtering should be done on a channel by channel basis, with some channels using stronger filtering than others. From initial testing, the lower frequencies are where most of the noise is contained, as well as low powered signals, such as the ones coming from the adjacent road around channel 1050.

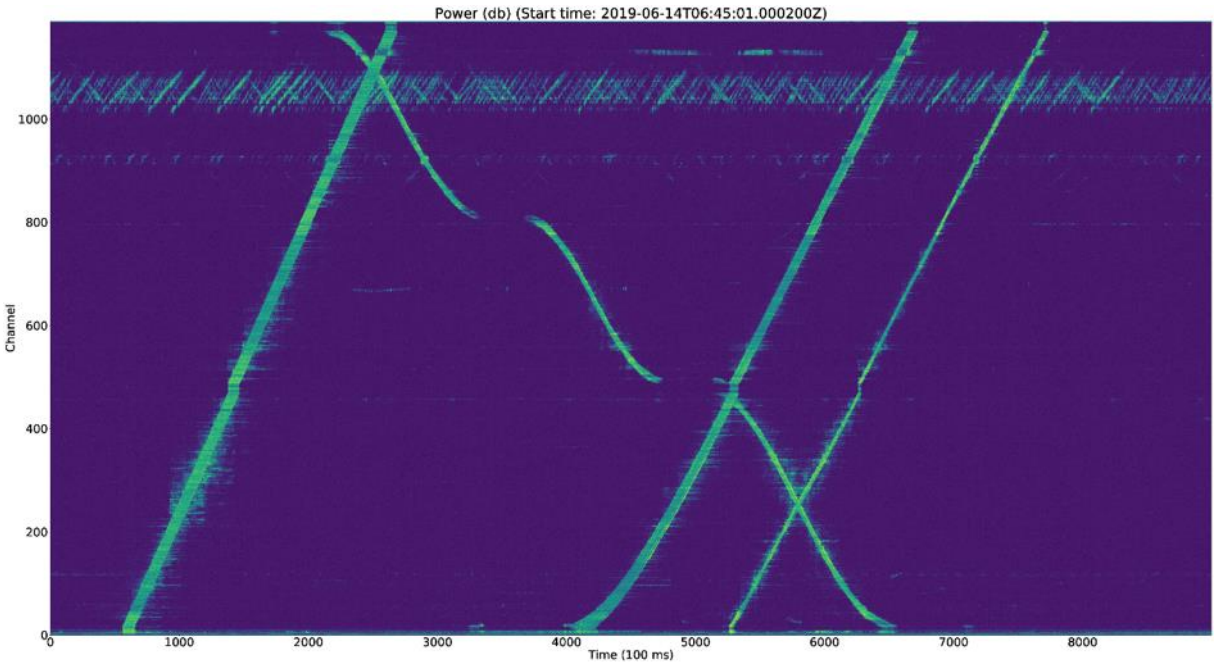

Figure 4-26: ESF Welch 500 (3 x 250 + 12) x 250 (hann) from 60 to 400 Hz

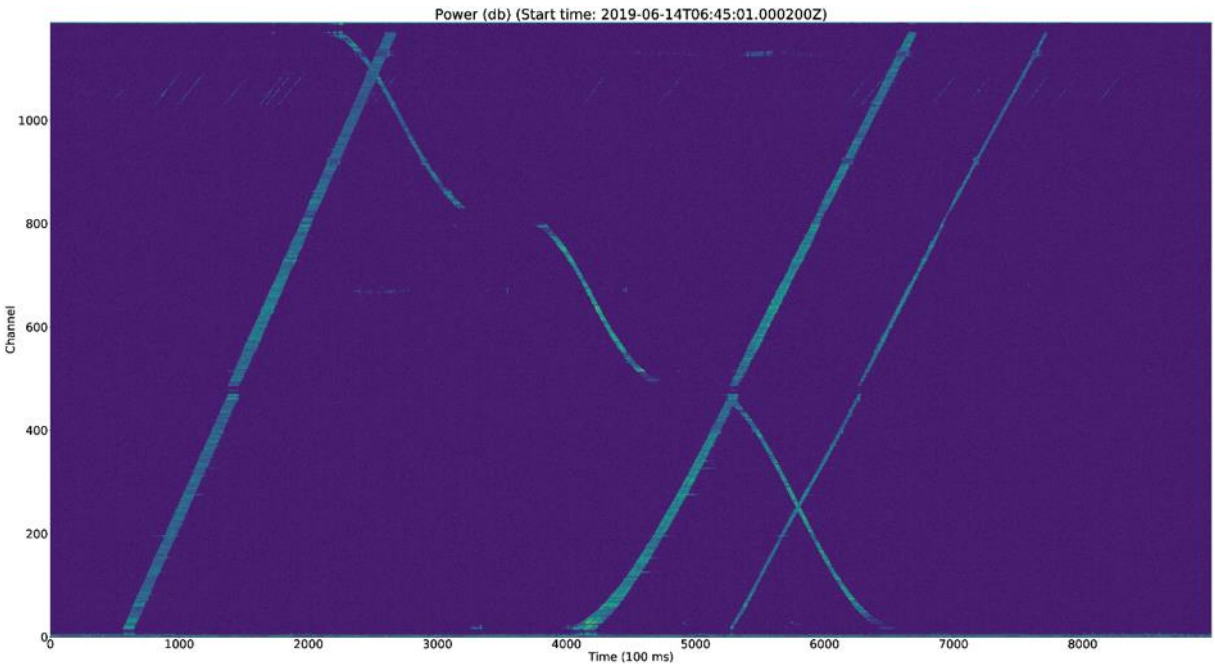

Figure 4-27: ESF Welch 500 (3 x 250 + 12) x 250 (hann) from 120 to 400 Hz

4.4.15 Thresholding

The result of the intra channel analysis is a binary value for each channel at each time step indicating if there is a train passing (1) or not (0) by each channel at this time. In a more sophisticated system, the threshold could in fact be a real number between 0 and 1 which encodes the confidence level that a train is passing by this channel at this time.

In any case, at some point during processing, a binary decision will have to be made. For this initial work, we have decided to output a binary value so that the inter channel analysis can decide to use it or not based on the physical properties of the train being tracked.

Also, at this point, no feedback from the inter channel to the intra channel has been implemented so that the thresholding is either fixed or dynamic, depending if the channel classification was done only once or is done continuously, respectively. Once again, at this point, we have only implemented a fixed thresholding based on a one-time channel classification of a time interval which should be representative of the train traffic in this segment.

4.4.15.1 Single Threshold

Single threshold is the simplest form of thresholding which is nothing more than the binary result of a comparison between a real value (signal) and a threshold value (threshold), returning a binary value in case the signal is above (1) or below (0) the threshold (or vice versa).

The main problem with single thresholding is when the signal oscillates between values above and below the threshold. In this case, the resulting binary value would also oscillate between 0 and 1, i.e., it will present high frequency noise.

This could be alleviated by using a nonlinear filter, e.g. a median filter, which introduces some delay (look into future values of the signal) but is able to eliminate isolated high frequency noise while maintaining the high frequency (edges) of the actual, long term signal. This would improve the results as long as there are few values above the threshold when there is no train activity and only a few values below the threshold when there is train activity. Longer lengths of the median filter would allow for the elimination of lower frequency at the expense of longer delays.

4.4.15.2 Hysteresis Threshold

Another thresholding scheme which presents some degree of nonlinear filtering in its implementation makes use of 2 threshold values, using the difference between these 2 values as hysteresis so that, for example, a value of 1 is generated when the signal is above the first threshold for the first time and for every next value which is also above the second threshold, which is smaller than the first. Once the signal is below the second threshold, a 0 will be generated for each next value until it finally exceeds the first threshold again, repeating the cycle.

It is easy to see that single thresholding can be viewed as a special case of hysteresis thresholding when both thresholds are the same.

4.4.16 Channel Classification

From the start, it was quite clear that some channels should not be used as they are sections of cable slack which are not laid out alongside the train tracks. Also, some sections present far too much noise and/or are buried in places that attenuate the signal to the point of making for a very low Signal to Noise Ratio (SNR), making it very difficult to discern signal from noise.

Also, as there are no baseline values for the noise, neither for the power spectrum nor the spectral flatness, there is not an absolute value which suffices for determining how much of each measure is an indication of the presence of a signal instead of silence.

However, as the Entropy Spectral Flatness (ESF) has a range that goes from 0 to 1, where 1 is produced by white noise and 0 when the signal is composed of a single sinusoid, its value can be used to determine how much filtering should be applied so that the silence periods produce a measure which is "close" to 1, i.e., are composed of white noise.

4.4.16.1 Classification heuristics

Given the 15-minute interval used, without any prior knowledge of the values and how they would compare with each other, and using the plot from Figure 4-8, we can see that:

1. There are 4 trains passing, 3 going from bottom to top without stopping and one coming from top to bottom stopping at 2 stations along the route. We assume that each train may produce signals with different energy levels.
2. Out of the approximately 9000 samples available, there are trains passing during approximately 300 samples for every channel, or 3.33% of the time.

Because there is only one train coming from top to bottom, which supposedly is on a different train track, and this train also stops at 2 stations, our assumptions do not hold for these 2 stations and a better sample (or longer one) should be chosen in order to improve the results.

Ideally, assuming the noise has no power and each train produces a constant power, which may be different for each one of them, we can devise a procedure to analyze each channel and produce a single threshold for each one by means of clustering. In this case, as there may be up to 5 different power levels, we could cluster the data into up to 5 clusters but could in fact use less clusters if 2 or more trains produce the same energy levels.

4.4.16.2 Clustering

Using a one-dimensional k-means clustering, we can partition the data into up to k clusters, each having a representative value which minimizes the total error for all the values. In the ideal case, the result would produce a perfect (0 error) fit and there would be a cluster for each different train energy level and one for the background, when there are no trains passing. In this case, the cluster values would be the exact energy levels of each train and the background (0).

In one dimension, the k-means problem can be solved for a global optimum with O(N) complexity, which is not the case with higher dimensions.

The objective function to minimize in the clustering process may not even be the total squared error across all partitions and it is possible that a different criterium may yield better results, however, for the example below, we have used the traditional squared error as the measure to minimize in the clustering process.

Based on the characteristics of the sample interval, we can use up to 5 clusters in order to be able to partition the data in a few clusters whose sum of elements should not exceed a small percentage of the total number of samples (around 3.33% for the sample interval in question) and a large number of elements in the last cluster.

4.4.16.3 Median Filter

A median filter of length N is a non-linear filter which produces as output the median value of the last N values, in the case of a real time filter.

A median filter is an example as a border preserving filter which is able to remove some high frequency isolated noise but keeps sharp borders between the absence and presence of a "long duration" signal.

The use of a median filter will introduce a delay which is equivalent to $\lfloor N/2 \rfloor$, e.g., a median filter with length $N = 5$ will introduce a delay of 2 samples which, in our case, is equivalent to 200 ms. Longer filters allow for better filtering but are dependent on how much delay is acceptable.

4.4.16.4 Example

For this example, we have used 5% as being the number of samples in the first clusters and 50% as being the number of samples in the last cluster, using up to 5 clusters. The channels for which this arrangement is not possible were classified as unusable and were automatically removed from the thresholding process.

The following steps were taken to prepare the data for clustering:

1. The Welch method was used to estimate the PSD for the 15-minute interval. A section length of 625 samples (250 ms) every 250 samples (100 ms) was used and 10 DFTs with a length of 256 samples (which results in a window shift of 41 samples or overlap of 215 samples) were averaged for each section using a Hann window. There were in total 8998 steps of 100 ms each.
2. The resulting PSD estimate was filtered by only keeping frequency bins from 9 to 41, inclusive, resulting in a bandpass filter from 87.9 Hz up to 400.4 Hz and the ESF was calculated for each channel for every 100 ms step.
3. A median filter of length 5 was applied to each channel, resulting in a delay of 200 ms.
4. A clustering was done for each channel for 1, 2, 3, 4, and 5 clusters. The clusters were searched so that the number of elements in the first consecutive clusters were less than or equal to 450 (5% of 8998) and the total number of elements in the last cluster was greater than or equal to 4499 (50% of 8998). The smallest number of clusters with the maximum number of elements less than or equal to 5% was chosen as the representative number of clusters.
5. A threshold was chosen for each channel based on the minimum value of the last partition to be included so that the total number of elements is still less than or equal to 5% of the total number.
6. During runtime, another median filter of length 5 is applied across all channels. This does not introduce any delay or shift as the interval for which the median is calculated is centered on the channel being filtered.

Using the previous assumptions, the following channels did not present any solutions and were automatically excluded based on the ESF:

0, 1, 2, 3, 5, 475, 479, 480, 483, 484, 1172, 1174, 1175, 1177, 1179, 1180, 1181, 1182, 1188, 1189.

The resulting thresholded data for the sample interval is shown in Figure 4-28.

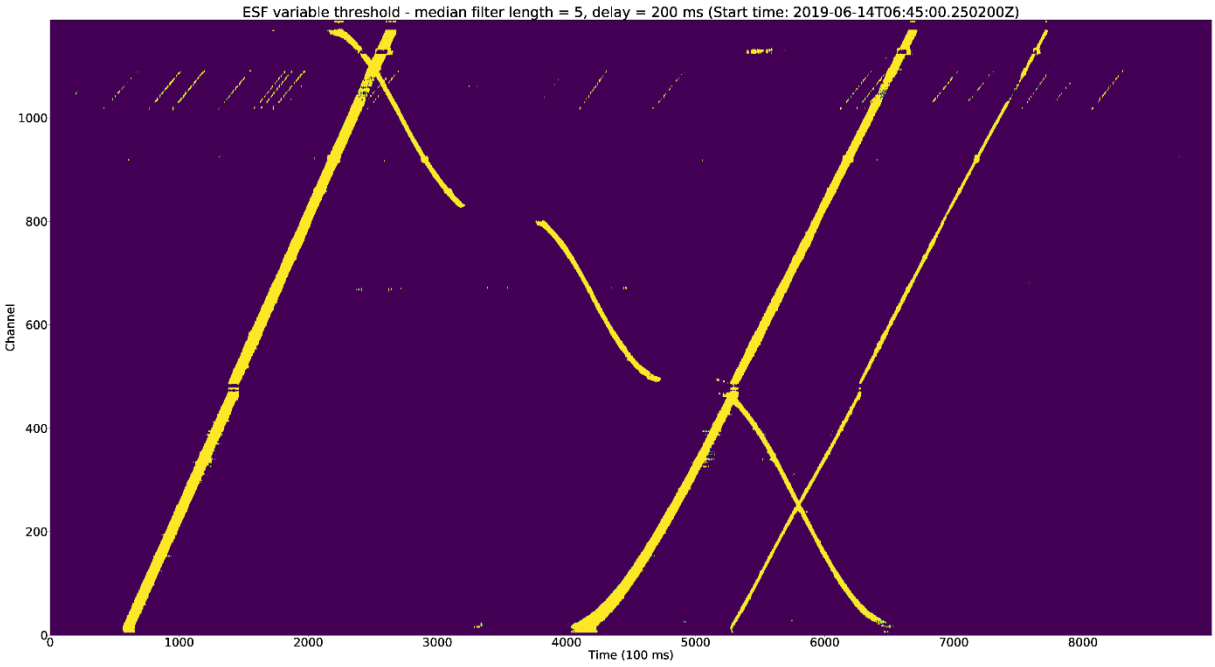

Figure 4-28: ESF threshold based on clustering

In order to test if the threshold values would hold for data outside the sample interval, the same thresholds were used for the whole 1 hour period, starting at exactly the same point as the sample interval. The resulting thresholded data is shown in Figure 4-29.

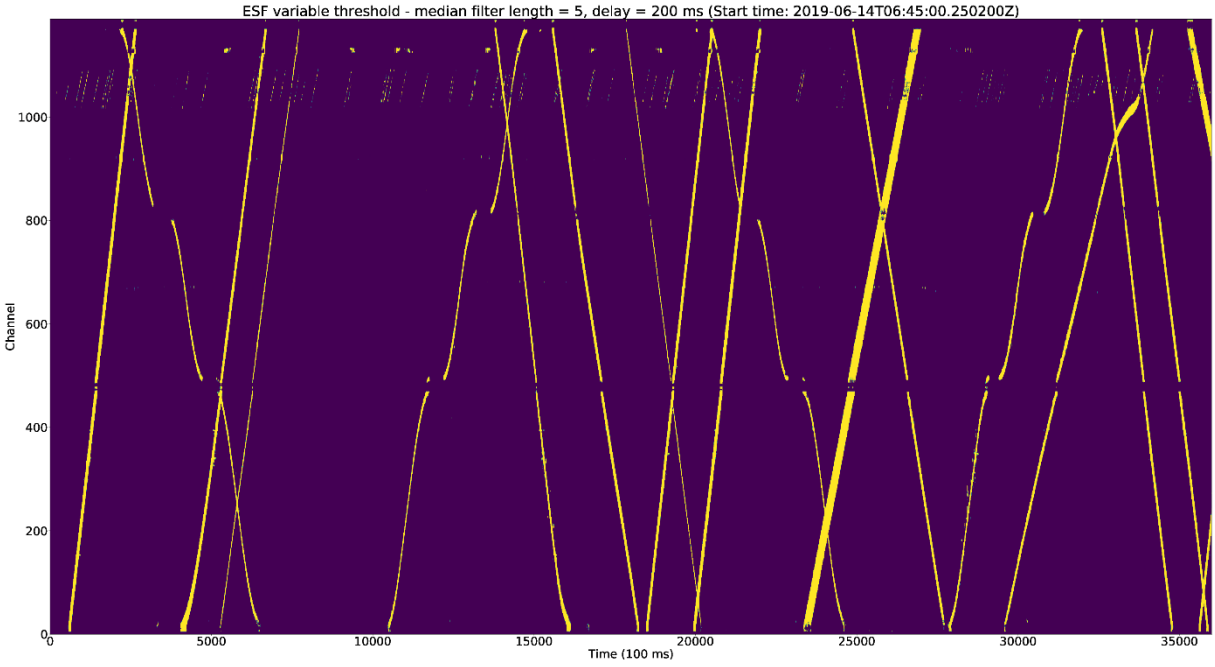

Figure 4-29: ESF threshold based on clustering for 1 hour period

This data was then used as input for the inter channel analysis. Of course, there is a huge number of parameters that can be changed in order to improve the threshold results.

In fact, both the power and the ESF can be used together in order to improve the results. Also, the clustering can be done with many different parameters and with different filtering and hysteresis thresholding could be implemented.

It is clear from these threshold plots that there is too much filtering for channels around channel 820 (Kissen station) which makes the trains on the track further away from the fiber to “disappear”. Also, filtering should be increased for the channels around channel 1050, where the highway is adjacent to the tracks in order to attenuate the weaker car signals.

Finally, Figure 4-30 shows the threshold results for the whole period from approximately 6:05 up to 13:00 from 14 June 2019 which is almost a 7 hour period.

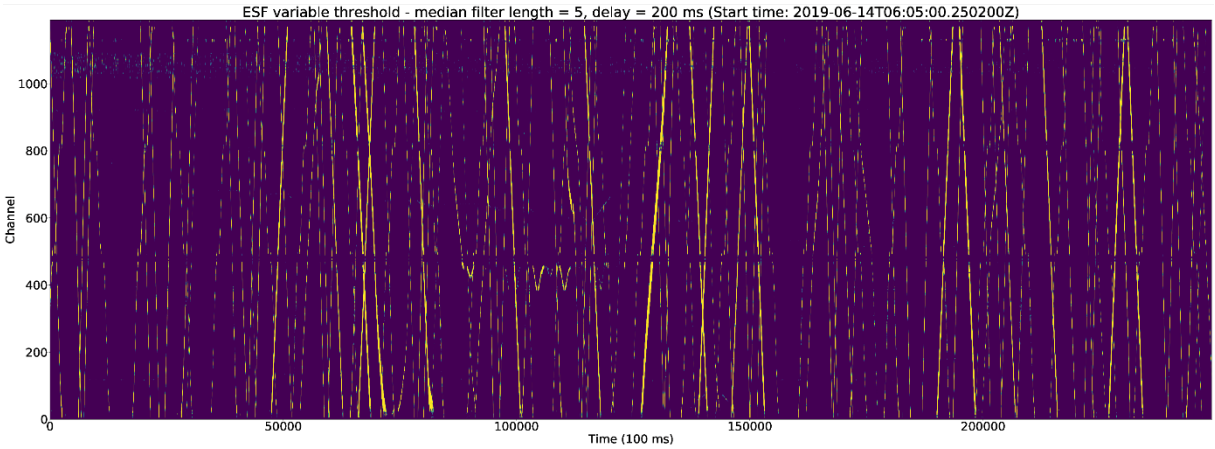

Figure 4-30: ESF threshold based on clustering from 6:05 up to 13:00

Further analysis of this data will be included in the intra channel analysis.

4.4.16.5 Cold Weather Data

We have received some more FOS data which was recorded on 15 December 2019, when the temperature was below freezing, in order to compare it with the one we already had, which was collected during summer time.

No calibration was done for this data and the same parameters which were used before were used to directly threshold this winter data including the exact same threshold values. The resulting threshold data is depicted in Figure 4-31.

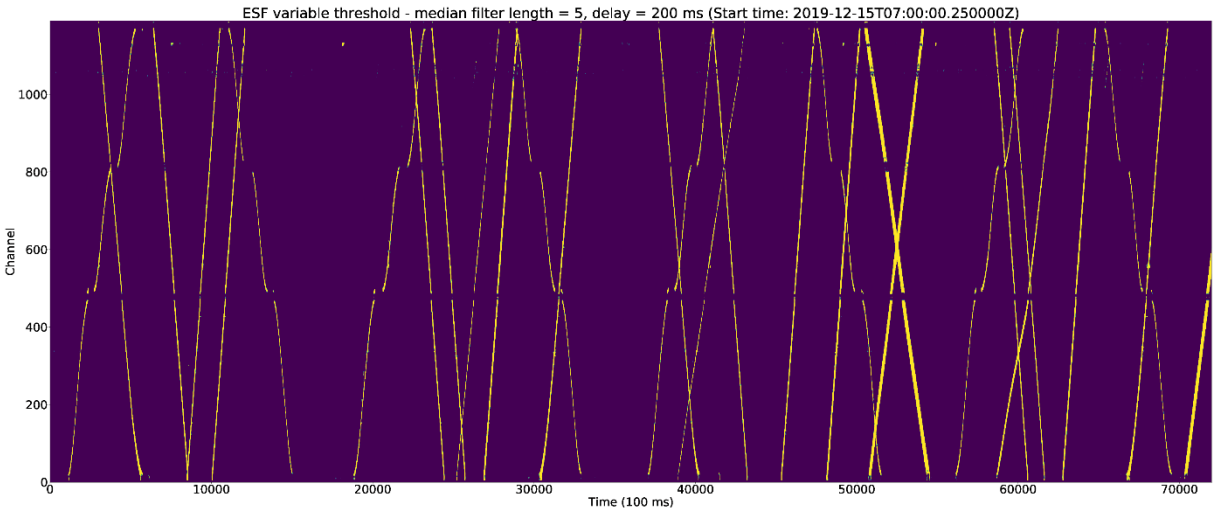

Figure 4-31: ESF threshold used on the cold weather data (2 hour period from 7 to 9 GMT on 15 Dec 2019)

From a visual inspection, this threshold data seems to be even better, showing less interference from the cars passing on the higher channels, than the one for the calibration period. This seems to indicate that the ESF measure is quite independent from the measured amplitude which does vary depending of the temperature.

4.4.17 Variable Filtering

As mentioned before, we have implemented variable filtering in order to try to eliminate more background noise and also try to minimize the effects of the signals coming from the cars passing by on the nearby road.

The filtering was done in an empirical way and, in fact, it has been increased, on average, to filter frequencies below 117.19 Hz and above 400.39 Hz for most channels except for the following, where in all cases the upper frequency is also 400.39 Hz:

- 420 - 459: 87.89 Hz
- 460 - 505: 58.59 Hz
- 506 – 525: 87.89 Hz
- 780 – 794: 87.89 Hz
- 795 – 825: 58.59 Hz
- 826 – 914: 87.89 Hz
- 926 – 1019: 87.89 Hz
- 1122 – 1134: 146.48 Hz

This was done in an attempt to do less filtering around the stations as the signal has been shown to be highly attenuated in this region.

Also, the channel classification took the one hour interval into consideration and not only the first 15 minutes as this shorter interval only contains a single train coming down which also stops at both stations. The one hour interval has more trains going up and down and contains trains that stop and don’t stop in both directions.

Just like before, we have used the same parameters to determine the single thresholds, which were chosen based on k-means clustering. The results are shown in the following figures.

Figure 4-32, Figure 4-33, Figure 4-34, and Figure 4-35 show the variable filter thresholding results for the 15-minute, 1-hour, and 7-hour intervals on 14 June 2019, and the 2 hour interval on 15 Dec 2019, respectively.

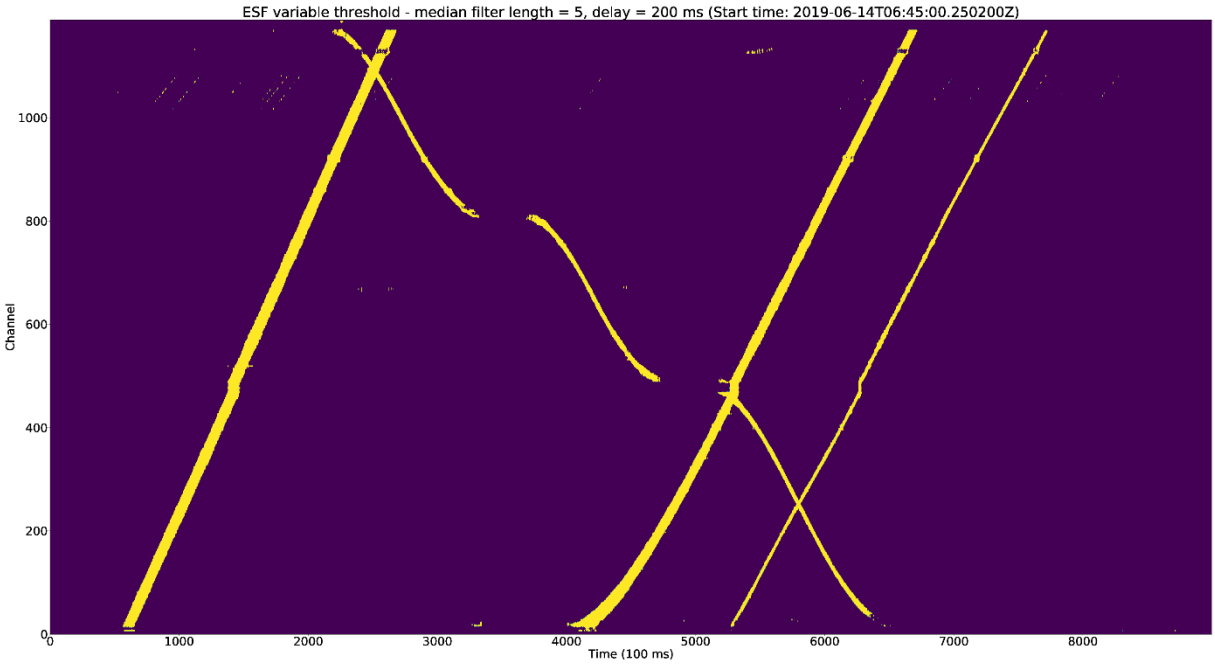

Figure 4-32: ESF threshold based on clustering with variable filtering for the sample 15 minute interval (14 June 2019)

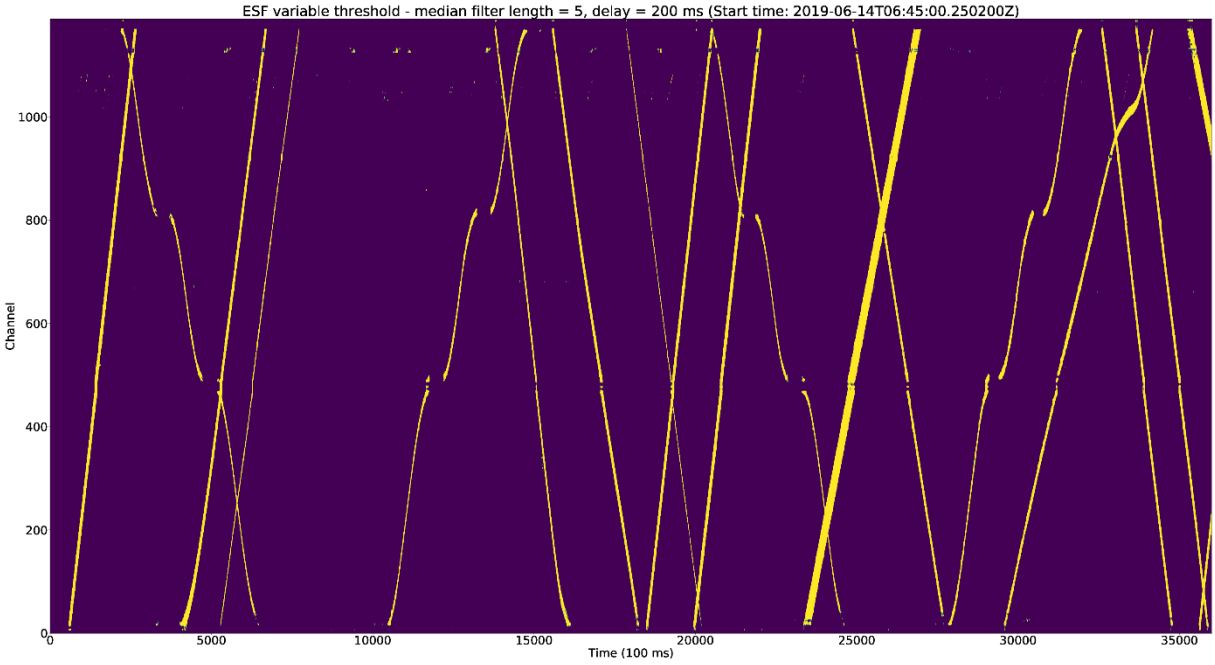

Figure 4-33: ESF threshold based on clustering with variable filtering for the 1 hour interval (14 June 2019)

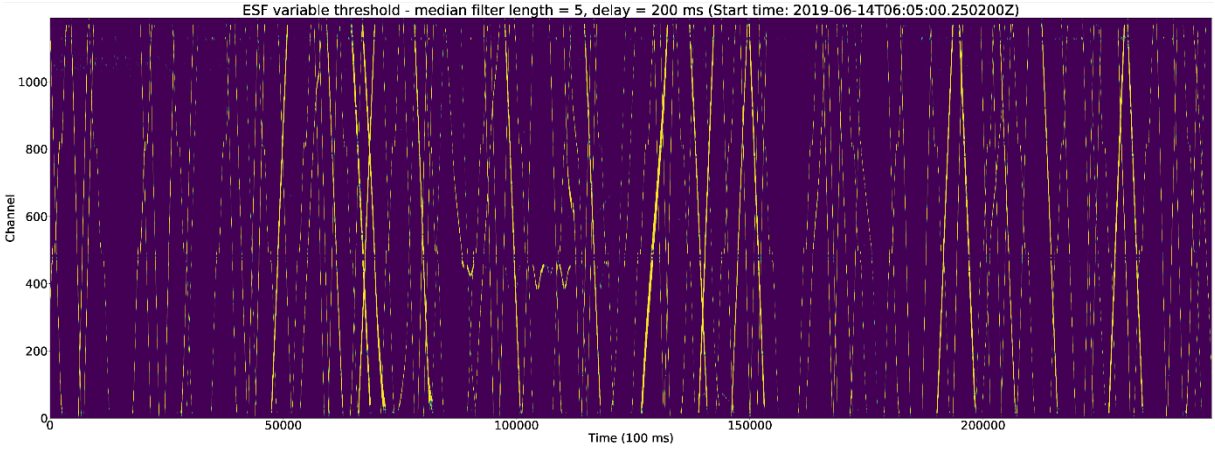

Figure 4-34: ESF threshold based on clustering with variable filtering from 6:05 up to 13:00 (14 June 2019)

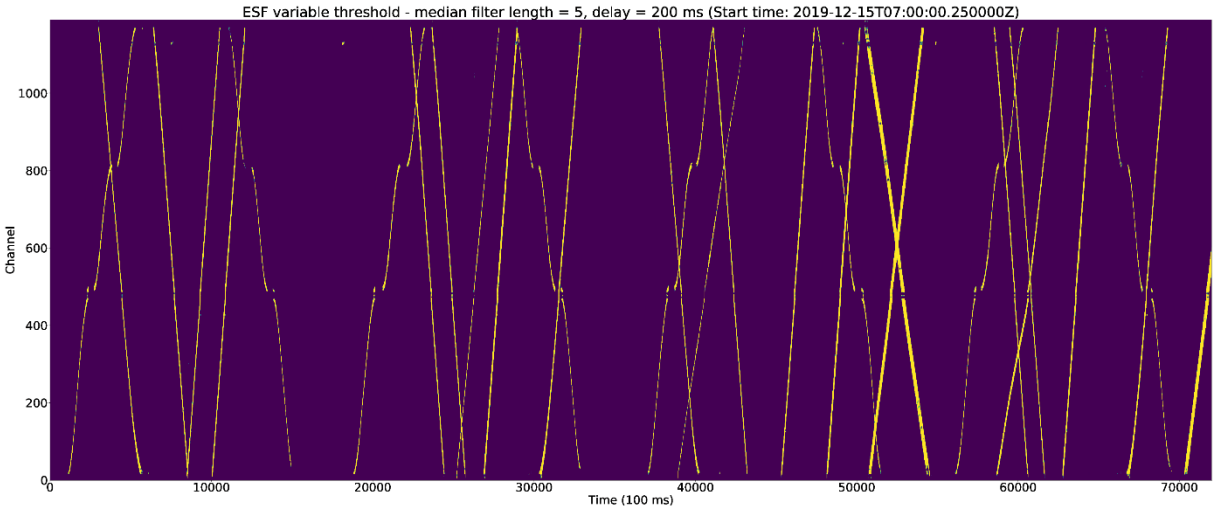

Figure 4-35: ESF threshold based on clustering with variable filtering for the cold weather data (2 hour period from 7 to 9 GMT on 15 Dec 2019)

For completeness, both the power and ESF (1-ESF) for the sample channels 190, 820, and 1050 are shown below, ready to be thresholded, i.e., with all filtering applied.

The graphs below take into consideration the filtering used for each channel which was:

- Channel 190: from 117.19 to 400.39 Hz
- Channel 820: from 58.59 to 400.39 Hz
- Channel 1050: from 117.19 to 400.39 Hz

All data has been median-filtered with a filter of length 5 and the figures show the values as they were used as input for the thresholding procedure. As a reminder, this introduces a 200 ms delay which is still well below the required minimum, which is 1 sec.

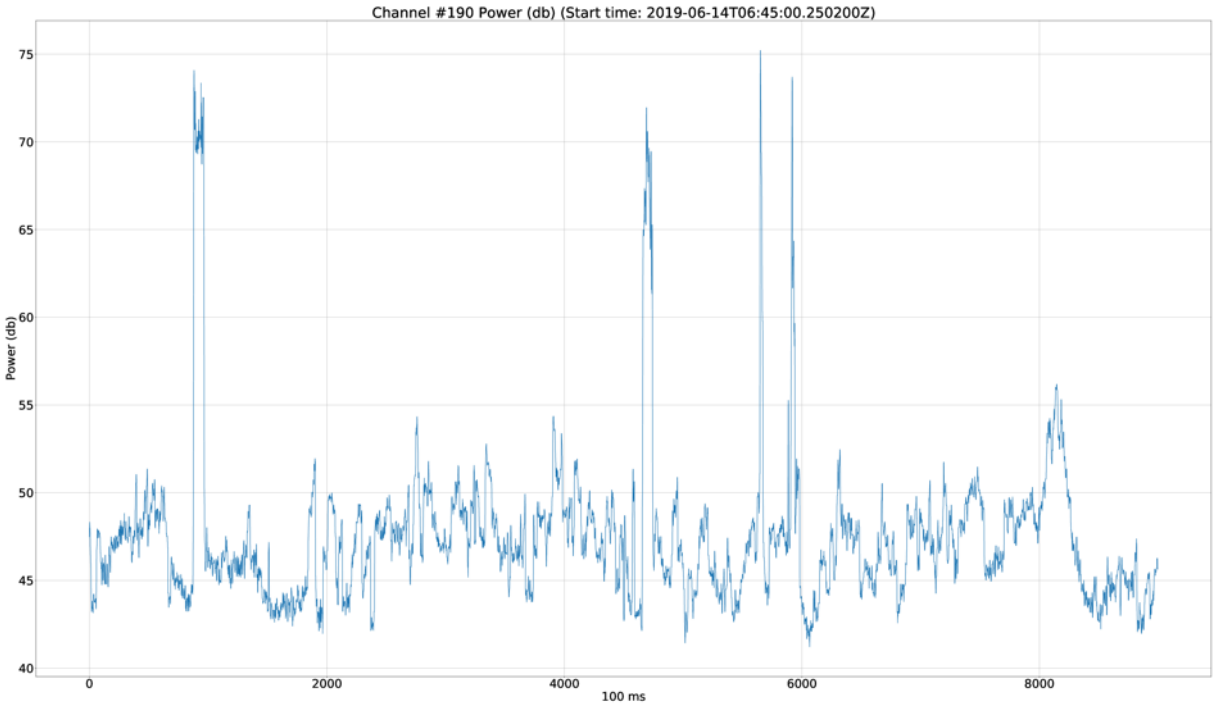

Figure 4-36: Channel 190 power (db) before thresholding

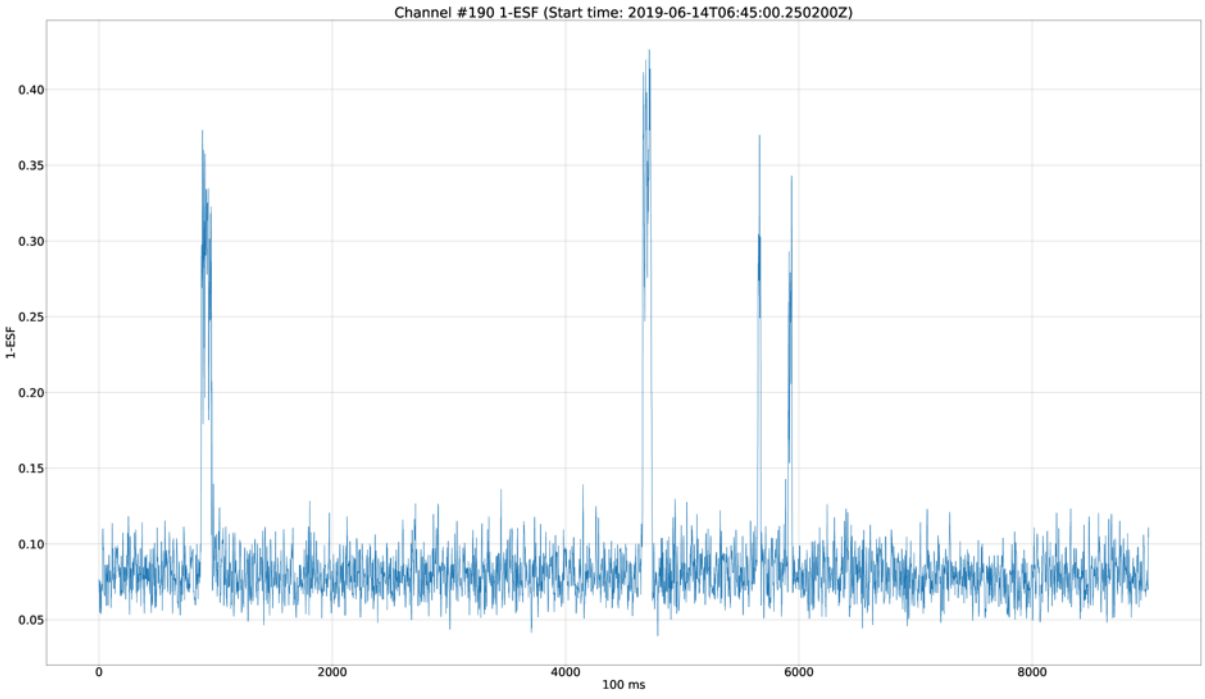

Figure 4-37: Channel 190 ESF (1-ESF) before thresholding

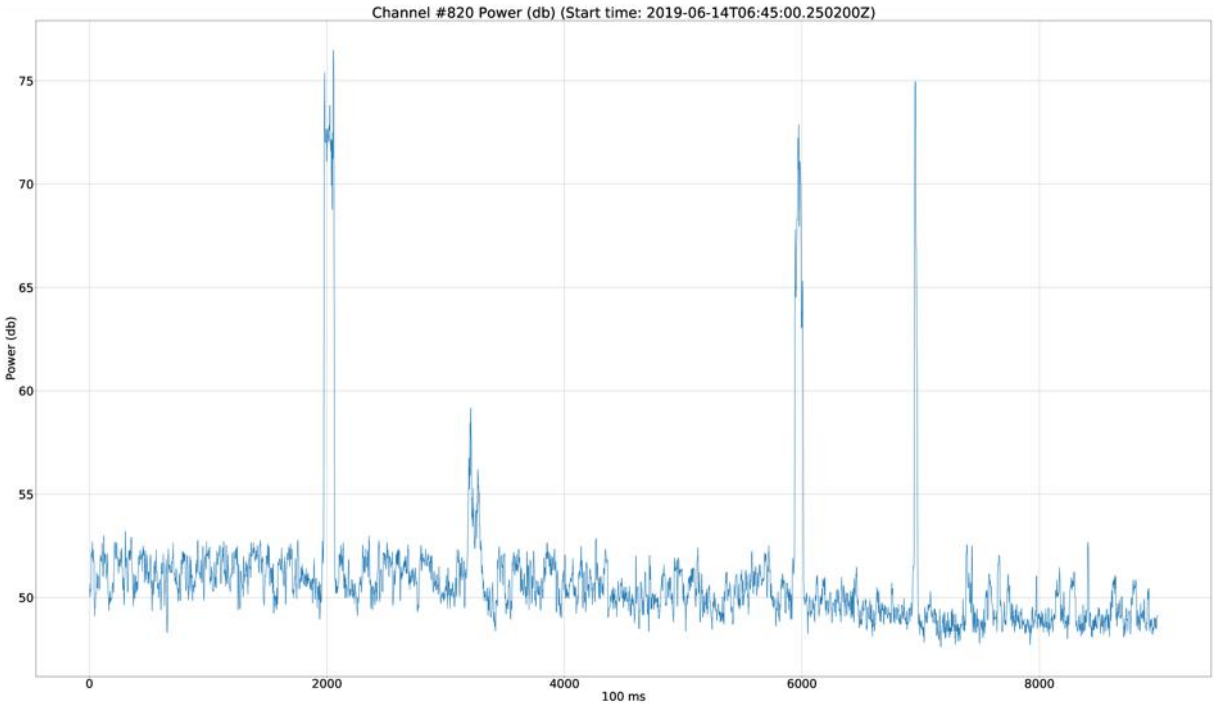

Figure 4-38: Channel 820 power (db) before thresholding

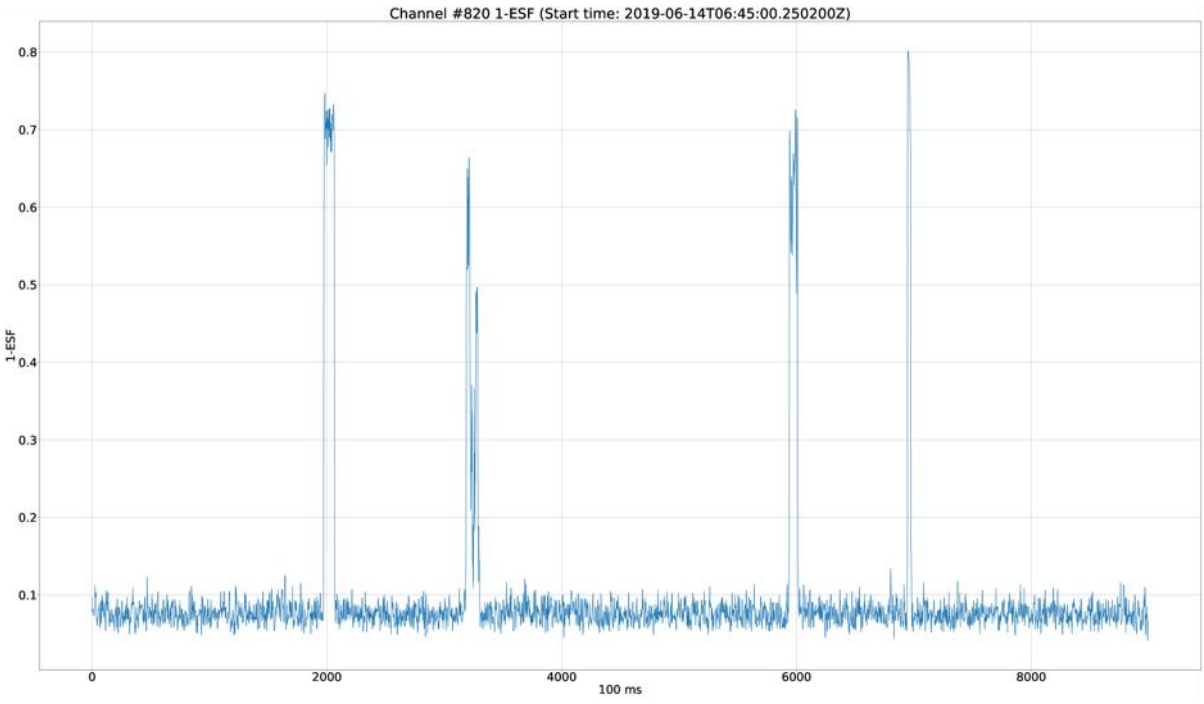

Figure 4-39: Channel 820 ESF (1-ESF) before thresholding

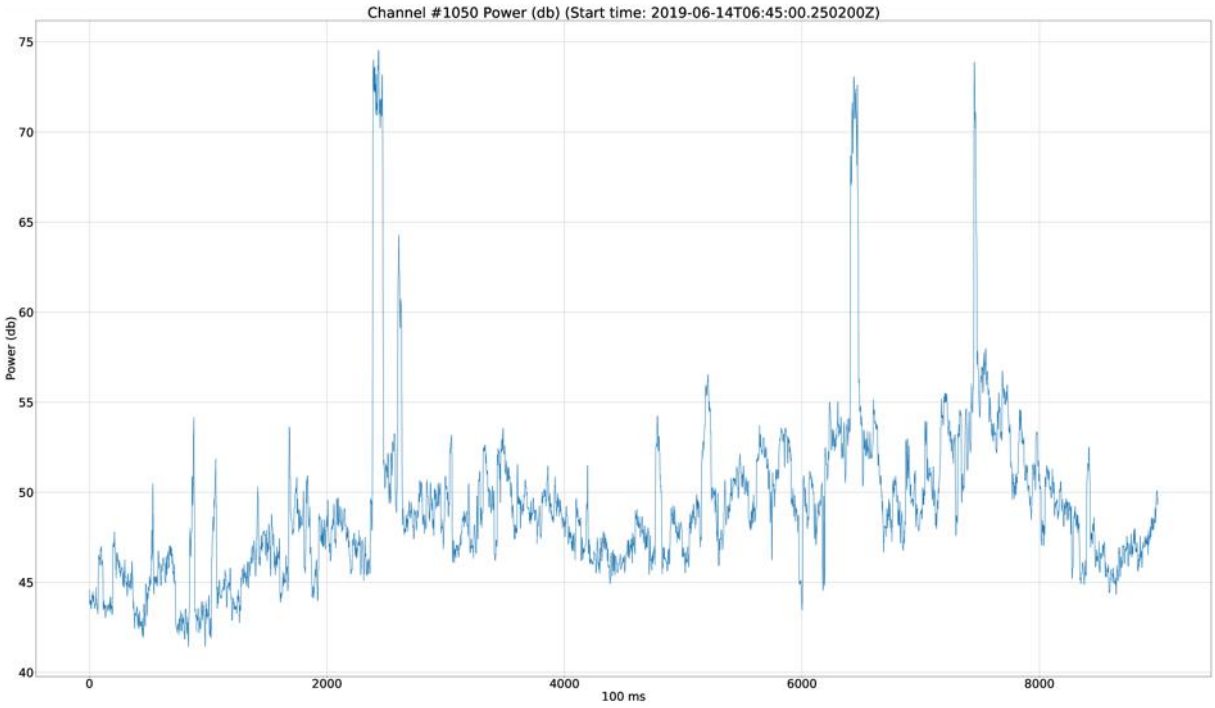

Figure 4-40: Channel 1050 power (db) before thresholding

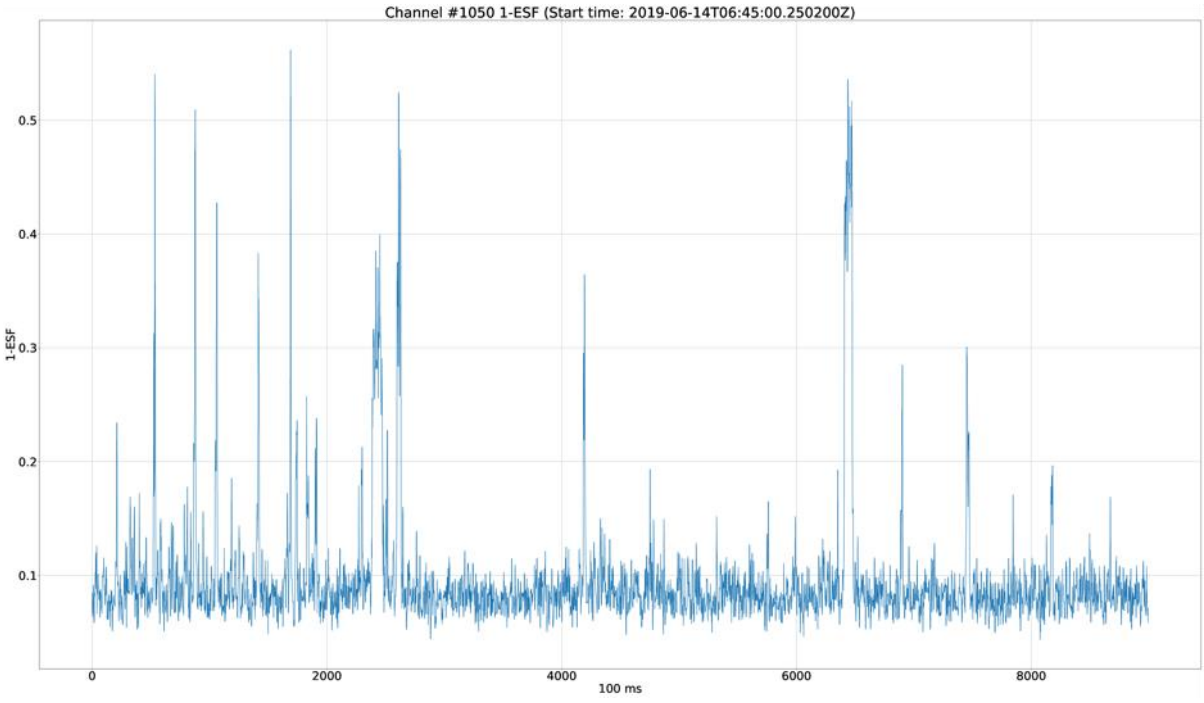

Figure 4-41: Channel 1050 ESF (1-ESF) before thresholding

Most channels present a better SNR using the ESF than using the Power measures, except for the ones which also contain signals coming from the cars in the section around channel 1050, which could be remedied up to a certain point by increasing the filtering.

It should be pointed out that the higher the filtering the shorter the train will show up for the inter channel analysis. This can and should be taken into account when designing the train tracking algorithms.

4.5 Inter Channel Analysis

4.5.1 Mapping Tables

In order to use FOS for absolute train localisation, it is necessary to map the fiber optic cable channels to real world coordinates on the track. This is needed because the cable is not always laid parallel to the tracks and because of cable slacks (rolled up lengths for maintenance reasons). In order to sort out this “unused” (rolled up) channels and to get exact positions for the remaining channels, a calibration was done on 23 October 2019 by hammering on the catenary masts along the track which led to a high amplitude changes in the actual channels of the FOS measurements. The channel with the highest amplitude was mapped to the coordinate of the respective mast. A catenary mast is located at approximately every 50m along the track.

The fiber optic cable in use is 445m longer than the track. At station Wichtrach, 147m of fiber optic cable are rolled up which was determined by analysing the energy spectrum shown in Figure 4-42. The same channels were also marked as “unused” in section 4.4.16.4. These channels were removed from the mapping. No other area of rolled up fiber optic cable was detected, which means that the remaining overlength is due to the non-parallel routing of the fiber optic cable. This remaining overlength was eliminated by linear interpolation of the channels between the catenary masts.

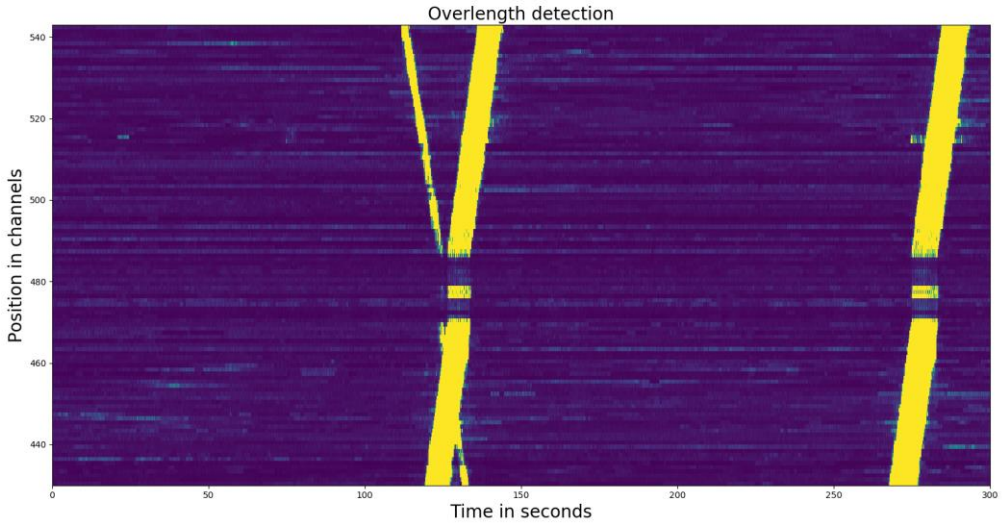

Figure 4-42 Energy spectrum at station Wichtrach showing the area of the rolled-up fiber.

The first step after receiving the complete mapping between track and channel was to move the catenary mast coordinates perpendicularly on the track with the use of OpenStreetMap [24]. As already mentioned, linear interpolation between the coordinates of the catenary masts was used to get the position for each channel between the catenary masts.

A table was finally obtained, which contains the mapping between fixed real-world coordinates (WGS84) for both tracks and their corresponding fiber optic cable channels from channel 36 to 1169. To facilitate the calculation of distance, speed, and train length and also to simplify the Kalman Filter model used in chapter 4.5.2, the mapping contains the track distance in metres, starting with the origin (0m) at channel 36. Table 4-1 shows a small extract of the mapping table.

Table 4-1 Small extract of the mapping table.

Channel	Latitude Track 1	Longitude Track 1	Latitude Track 2	Longitude Track 2	Distance Track 1	Distance Track 2
36	46.872227	7.559690	46.872218	7.559631	0.000000	0.000000
37	46.872153	7.559711	46.872145	7.559653	8.289265	8.289259
38	46.872080	7.559733	46.872072	7.559674	16.578531	16.578518
39	46.872007	7.559754	46.871999	7.559696	24.867796	24.867778

Figure 4-43 shows the resulting mapping at station Wichtrach. The catenary masts are marked with white circles with their corresponding channel number. The green and blue circles mark the position of the channels on the two tracks.

Trains in Switzerland generally drive on the left-hand side, however, there can be exceptions. Currently, FOS is not track selective, i.e., it cannot differentiate between the tracks in which the trains are running. Thus, the regular driving direction is assumed. Trains from Münsingen to Uttigen are driving on track 1, i.e. in increasing channel order and trains from Uttigen to Münsingen are driving on track 2, i.e. in decreasing channel order. The fiber is laid closer to track 1.

Figure 4-43 Google Earth picture showing the resulting mapping at station Wichtrach. Catenary masts are marked with white circles with their corresponding fiber channel. The green and blue circles mark the channel position on the track.

Figure 4-44 shows two areas where FOS can be affected by external influences. Cars passing on the highway or bridges with strong vibrations when trains pass by them.

Figure 4-44 Google Earth picture showing the mapping where FOS can be affected by external influences. Cars driving on the highway produce noise and the whole bridge vibrates when a train is passing which makes it more difficult to get a clear front and rear end at this section.

4.5.2 Train Position Prediction

Each channel of the fiber optic cable has slightly different characteristics and, therefore, noise and signal levels are different from channel to channel. This results in slightly different front and rear end detections for each channel in the intra channel analysis, which makes it necessary to use a Kalman Filter in the inter channel analysis to improve the robustness of the train position detection.

At first, the quantities of interest which are to be refined (filtered and predicted) by the Kalman Filter are defined. For train localisation with FOS, the position of the front and rear ends, as well as the speed and the length of each train are of high interest. This results in the state vector as follows:

$$\boldsymbol{x}_k = \begin{bmatrix} s_{f,k} \\ s_{r,k} \\ v_k \\ a_k \\ L_k \end{bmatrix}$$

where $s_{f,k}$ is the position of the front end and $s_{r,k}$ is the position of the rear end of the train along the track. Please note that the front end is assumed to be the position closer to the origin and is therefore not the real front end in the classical sense. v_k is the speed, a_k is the acceleration in the driving direction, and L_k is the train length. Figure 4-45 depicts these values visually. The acceleration vector points in the same direction as the speed vector.

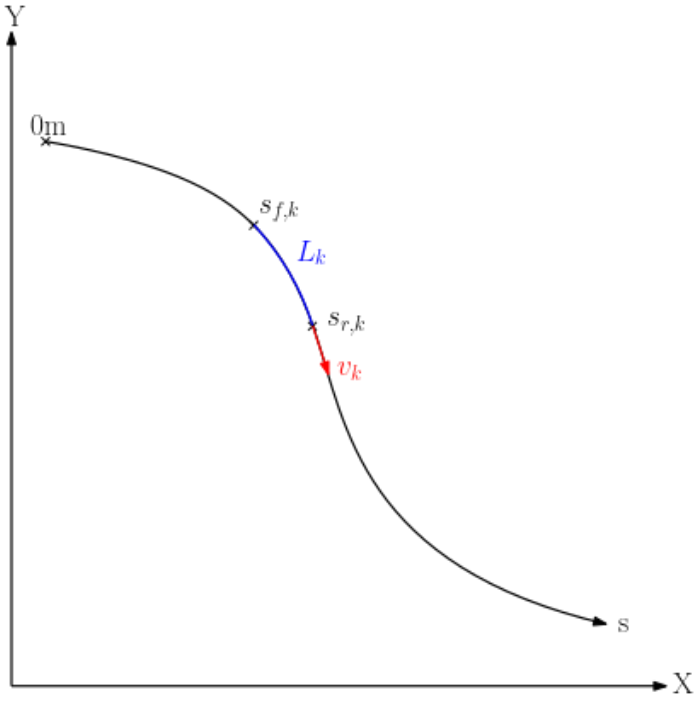

Figure 4-45 Illustration of the definition of the state vector. X and Y indicate the real-world coordinates, 0m indicates the origin at channel 36.

The state vector is initialized when the algorithm starts tracking the train and some measurements are already available to estimate all the state values.

A train has limited acceleration and deceleration. Thus we use a constant acceleration model for the state prediction:

$$\mathrm{x_k} = \mathrm{F_k x_{k-1}} + \mathrm{w_k}$$

With

$$F_k = \begin{bmatrix} 1 & 0 & T & T^2/2 & 0 \\ 1 & 0 & T & T^2/2 & 1 \\ 0 & 0 & 1 & T & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

being the state transition matrix, $\mathrm{w_k}$ the process noise, and T the sampling time.

The measurements can be written as

$$z_k = H_k x_k + \nu_k$$

with

$$H_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

being the measurement matrix and ν_k the measurement noise.

Both the process noise $\mathrm{w_k}$ and the measurement noise ν_k are assumed to be modelled as a zero mean white noise with process covariance Q_k and measurement covariance R_k.

They are defined as the expected values of the process noise and measurement noise vectors:

$$Q_k = \mathrm{E}\{w_k w_k^T\} = \begin{bmatrix} \sigma_{sh} & 0 & 0 & 0 & 0 \\ 0 & \sigma_{st} & 0 & 0 & 0 \\ 0 & 0 & \sigma_v & 0 & 0 \\ 0 & 0 & 0 & \sigma_a & 0 \\ 0 & 0 & 0 & 0 & \sigma_L \end{bmatrix}$$

$$R_k = \mathrm{E}\{n_k n_k^T\} = \begin{bmatrix} \sigma_h & 0 \\ 0 & \sigma_t \end{bmatrix}$$

The values R_k and Q_k were determined by trial and error. This can be done because the actual values of R_k and Q_k are not very important. More important is the difference between the two values. Small values of Q_k and big values of R_k mean good filtering of the measurements. The opposite would be a highly dynamic model. Because of the high inertia of trains, the values for Q_k and R_k were chosen for good filtering of the measurements. The input for the Kalman Filter comes from the Intra Channel Analysis and can be seen as a contour plot in Figure 4-46. On the horizontal axis the time is plotted and on the vertical axis the position along the track with the origin at channel 36. The green lines represent the transition between noise and signal.

When looking at the first train which starts at second 70 and goes from bottom to top, the lower edge is defined as the front end of the train s_f and the upper edge as the rear end of the train s_r. The difference between front and rear end is the train length L and the slope of the two lines is the train speed.

The front and rear ends of the four trains can be seen clearly as well as noise areas. Just around channel 1000 some short movements can be seen, which are the cars travelling on the highway near the track and they add additional measurement errors. Also, around channel 1050 the measurements get poor because of the bridge, which is constantly vibrating when a train passes by it. But these errors will be removed in the Inter Channel Analysis.

For the sake of completeness, the values used for R_k and Q_k are: $\sigma_h = \sigma_t = 500$, $\sigma_{sh} = \sigma_{st} = 0.05$, $\sigma_v = 0.01$, $\sigma_a = 0.001$, $\sigma_L = 0.1$.

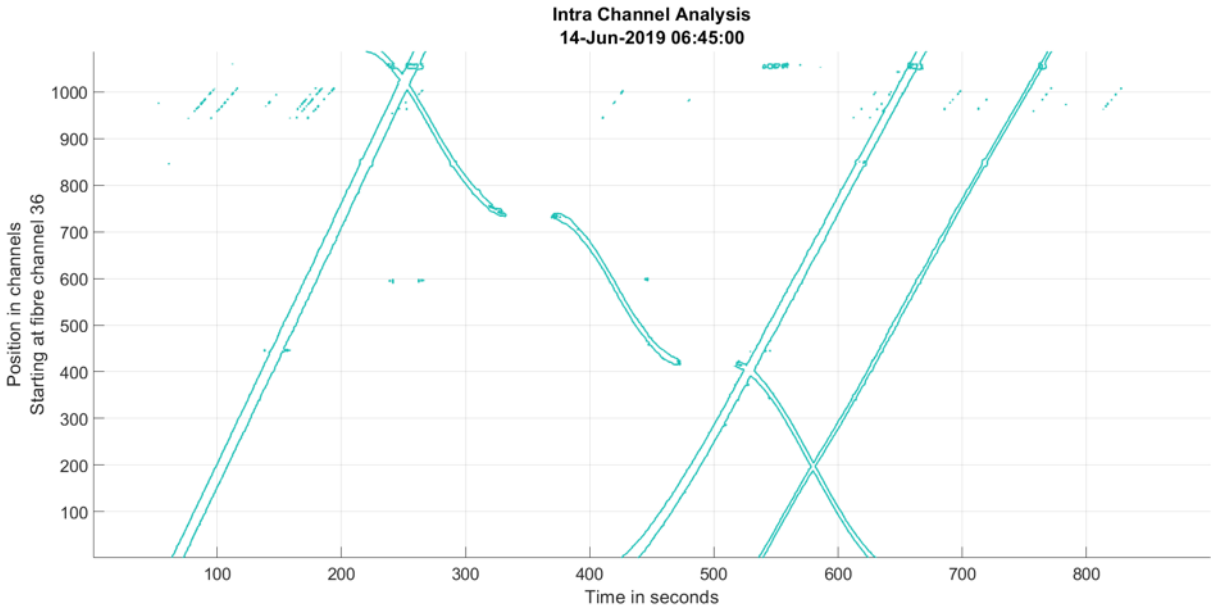

Figure 4-46 Results of the Intra Channel Analysis represented as a contour plot. The green lines show the transition between noise and signal.

When a new train is detected it will be tracked by the algorithm. After an initialization phase, the Kalman Filter will be initialized by estimating the state vector x_0 from the previous measurements and the state covariance P_0 is set to I, which is the identity matrix. Every train moving on the track has its own Kalman Filter, which does not change during the whole tracking. The initialization phase depends on the train length and on the measurement quality, because the state vector can only be initialized when the train is fully present (front and rear end of train) in the measurement area.

For each new measurement assigned to a specific train, the Kalman Filter process is as follows [25]:

1. Prediction of the state
$$\hat{x}_k = F_k x_{k-1}$$
2. Prediction of the state covariance
$$\hat{P}_k = F_k P_{k-1} F_k^T + Q_k$$
3. Calculation of the innovation covariance
$$S_k = H_k \hat{P}_k H_k^T + R_k$$
4. Calculation of the Kalman Filter gain
$$K_k = \hat{P}_k H_k^T S_k^{-1}$$
5. Update state estimation with the measurements
$$x_k = \hat{x}_k + K_k(z_k - H_k \hat{x}_k)$$
6. Update state covariance
$$P_k = (I - K_k H_k)\hat{P}_k$$

The Kalman Filter also plays an important role when trains going in opposite directions cross. Within this time interval there is only one front and one rear end measurement for both trains and the Kalman Filter is used to predict the state x_k for this short time interval.

4.5.3 Train Tracking

Applying the inter channel analysis algorithm, trains can be easily tracked on their way along the track. Thus, FOS could be used as an additional sensor for absolute localisation.

The inter channel analysis uses the results from the intra channel analysis as input data. Different approaches in the intra channel analysis are compared here and the best one will be chosen for the final evaluation with the axle counter data as ground truth.

For the first evaluation 15 minutes of the data from the measurement day on 14 June 2019 were used. Figure 4-47 shows the results from the Intra Channel Analysis as a contour plot in green. The red lines are the calculated transitions between noise and signal from the Inter Channel Analysis. It can be seen that the four trains were tracked very well. It can be seen that the Kalman Filter does a god job in predicting the train position even for the train that going from top to bottom and stops at both stations. When a train stops it does not produce any vibrations and cannot be “seen” with FOS. Later in section 4.5.5 it will be investigated at which speed a train gets “lost” by FOS.

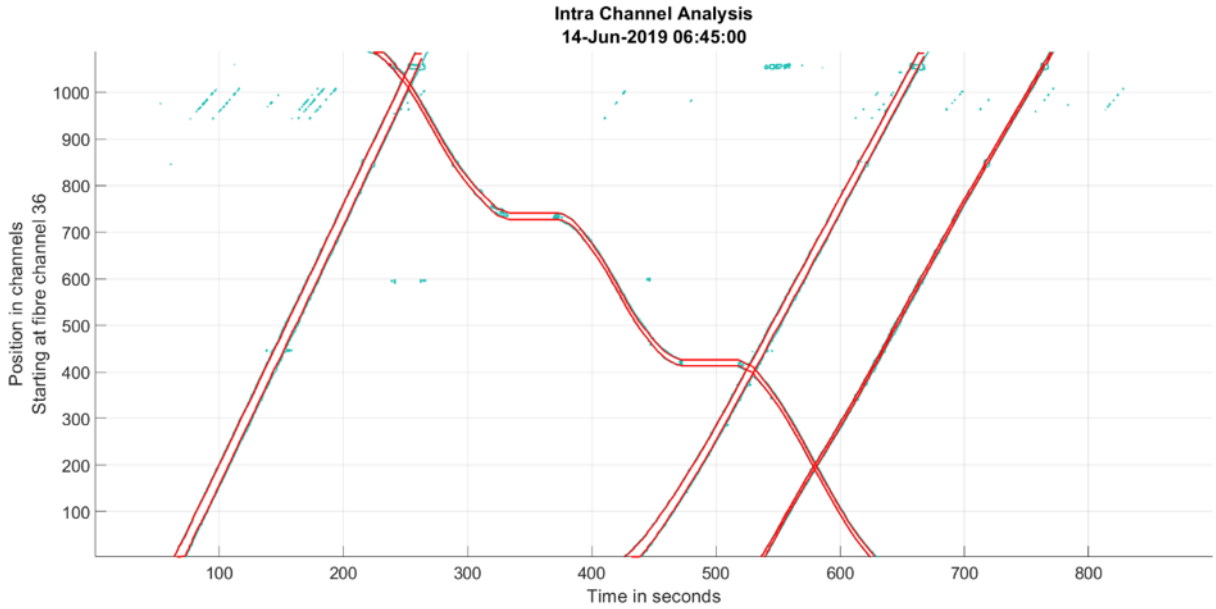

Figure 4-47 Comparison between Intra and Inter Channel Analysis in a contour plot. The green lines represent the transition between noise and signal and the red lines show the achieved tracking of trains.

Figure 4-48: shows four plots with different approaches and parameters used in the Intra Channel Analysis and the corresponding results achieved in the Inter Channel Analysis. The parameters used can be found in the title of each plot. First of all, the method applied is mentioned, whereby we distinguish between Power and ESF Thresholding. For more detailed information, please refer to chapters 4.4.12 and 4.4.14. Secondly, the used threshold for deciding between noise and signal is mentioned, followed by the used frequency band. The delay specified at the end results from the use of a median filter. When using the power for signal detection it can be seen that the cars can be filtered well, which is not the case when using ESF for the detection. Later on, in section 4.5.5, it will be shown that using ESF leads to more precise length estimations, especially for the trains going from top to bottom. They are located further away from the cable and that's why they are generally determined shorter than trains on the track closer to the fiber.

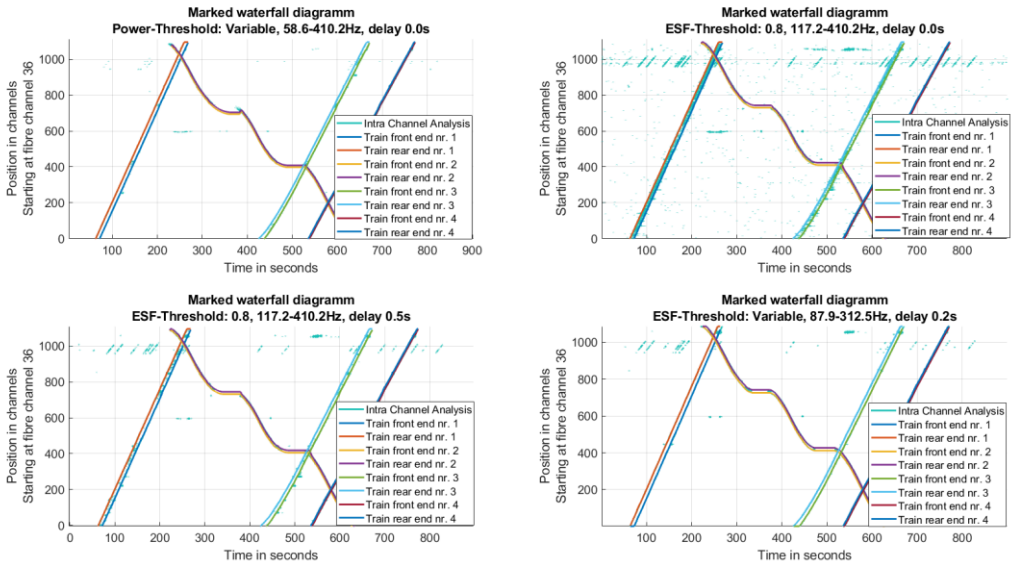

Figure 4-48: Different approaches and parameters in the Intra Channel Analysis and the calculated tracking of front and rear ends by the Inter Channel Analysis

The use of a median filter improves the data, as can be seen when comparing top right and bottom left plot. There are less disturbances (False Positive detections) in exchange for a longer time delay. However, considering that GNSS, balises and axle counters all work with the accuracy of seconds, the delay of 0.2s is definitely acceptable (a median filter with a length of 5 samples introduces a delay of 2 samples and each sample is 100ms in this example).

The last parameter set in the bottom right plot also uses a median filter along channels. This makes the edges of the train more accurate, but at the same time the disturbances get more visible. This filtering, however, should be applied after the mapping table is used to convert from channel number to linear track distance. In this example, this second median filter was applied directly to the channel numbers but this should be revised.

Sections 4.5.4 and 4.5.5 compare the proposed methods and parameters in order to identify the “best” among them.

4.5.4 Train Speed

The train instantaneous speed, which is defined as the rate of change of its position in relation to time, is another important quantity that needs to be estimated.

One possibility to calculate the train speed is to calculate the slopes of the front and rear end (which may not be the same specially when the train has non zero acceleration). This only needs to be done when initializing the Kalman Filter. Afterwards the speed is available from the state vector defined in section 4.5.2 anyway.

All methods and parameters of Intra Channel Analysis, which are compared in the following, achieved very similar results in speed estimation.

Figure 4-49 shows the speed of the four trains for the settings shown in bottom right plot of Figure 4-48:. As discussed in section 4.5.5, the best results were achieved using these settings. Unfortunately, there is no ground truth for the speed of all tracked trains measured with FOS on 14 June 2019, so no meaningful results can be presented in this section. But the tracking contains 9 runs with the measurement train (mewa 12) which was equipped with GNSS. The comparison with GNSS is shown in section 6.3.4, where the speed is also evaluated.

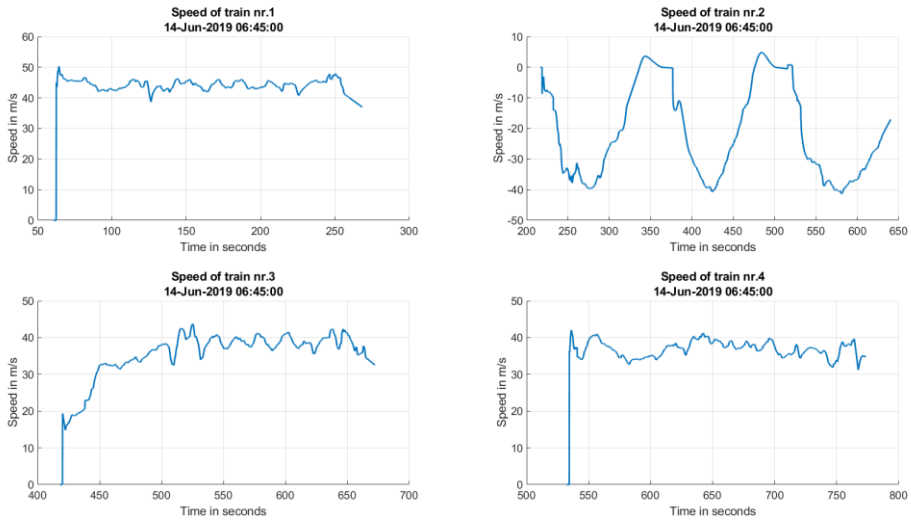

Figure 4-49 Speed of the four trains shown in bottom right plot of Figure 4-48:.

4.5.5 Front and Rear End Determination

Another important parameter for train integrity determination is the train length. Being able to measure the length of the train continuously along its way on the track, train integrity could be easily determined at any moment. In this section it is investigated how to use FOS for train integrity.

The train length can be derived from the difference between front and rear end and, just like the speed, is a variable in the state vector defined in section 4.5.2.

For all the different methods and parameters in the Intra Channel Analysis, Figure 4-50 shows the corresponding lengths to the tracking in.

Regarding the calculated lengths for trains 1, 3 and 4, it can be seen that the trains are estimated to be longer than they actually are. This can be explained by the fact that the vibrations of a train are measured even before the train arrives at this channel and continues after it leaves. However, it will be shown later that this length offset can be corrected. Train 2, on the other hand, is only estimated longer than it really is by using the ESF with variable threshold and median filter along the channels. Since train 2 runs on the track further away from the fiber cable, it seems that this method is more robust against the distance to the fiber optic cable.

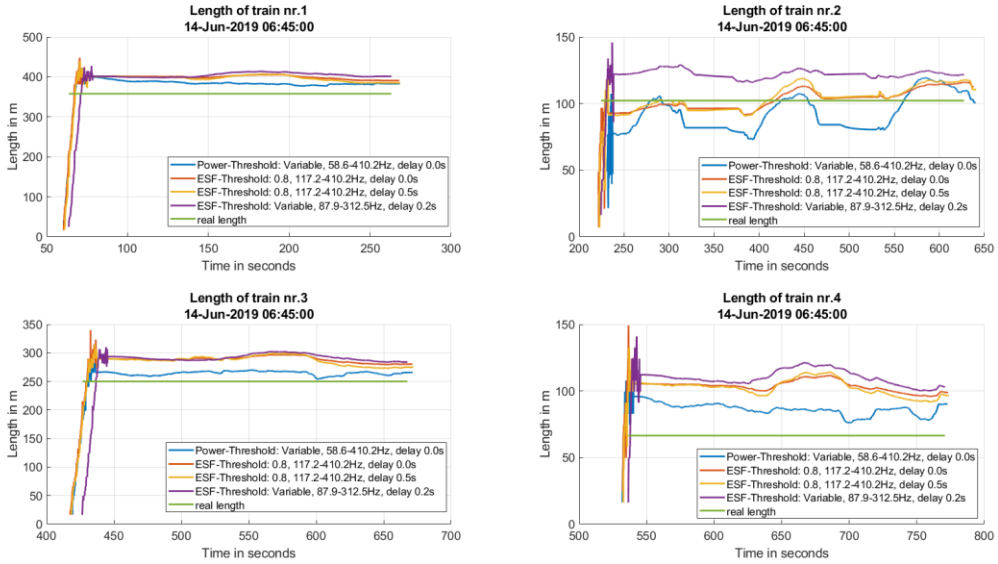

Figure 4-50 Lengths of the four trains compared for all the settings in the Intra Channel Analysis. Lengths match with Tracking in Figure 4-48:.

According to section 4.5.3 the power method with variable threshold had the advantage of a good filtering of the cars on the highway. In Figure 4-50 it can be seen that this method is highly dependent on the vibration intensity. The calculated length becomes more inaccurate with smaller trains (train 4) and greater distance to the fiber optic cable (train 2).

In the end, the ESF method with variable threshold and a median filter along the channels proves to be the best choice.

Another striking feature is the variation in the length of train 2, which occurs for all methods. Due to their similarity with the train speed, the correlation between length and speed was calculated and it was determined to be a linear dependency. This can be explained by the process of PSD estimation in the Intra Channel Analysis. The PSD was calculated using a 625-sample window. This means that in the worst case the detection of the front and rear ends could be shifted by this value. For a sampling rate of the interrogator unit of 2500Hz this would correspond to a delay of 625/2500Hz = 0.25s, which could explain the dependency on the speed.

Furthermore, the train is stretched or compressed during acceleration or braking due to the degrees of freedom in the couplings.

In order to get the correlation parameters, a simple optimisation problem was solved. Therefore, the tracking algorithm was applied to a data set that covered a period of 1 hour. Within this time period, 17 trains were tracked with different lengths and speeds. The optimisation problem was chosen to minimize the error between the real and calculated lengths:

$$\min_{g} \sum_{i=1}^{N} |e_i|$$

with the error defined as

$$e = l_r - (l_c - g_1|v|)g_2$$

where *i* stands for the evaluation at *N* different time points. l_r is the real length, l_c is the preliminary calculated length (rear end position minus front end position), v is the train speed and g_1 and g_2 are the optimisation parameters. The linear correlation with the speed shall be described by g_1 and a possible offset should be eliminated by g_2.

The optimisation problem was solved separately for each track because of the dependency on the distance between the track and the fiber. Even though the intra channel analysis using the ESF with variable threshold is more robust in relation to the track distance, this dependency still exists.

The corrected train lengths for the four trains currently under consideration can be seen in Figure 4-51 as red curves. These are the curves after the tracking with the Kalman Filter and the applied correction from the optimisation. The blue curves show the calculated train lengths after the intra channel analysis only. As expected, the inter channel analysis and correction results in a significant improvement. Please note that the train length correction was not yet considered in Figure 4-47 and Figure 4-48:, but it will be for all following results. The calculated length after the Intra Channel Analysis also corroborates the choice of the parameters Q_k and R_k in the direction of better filtering instead of more dynamics.

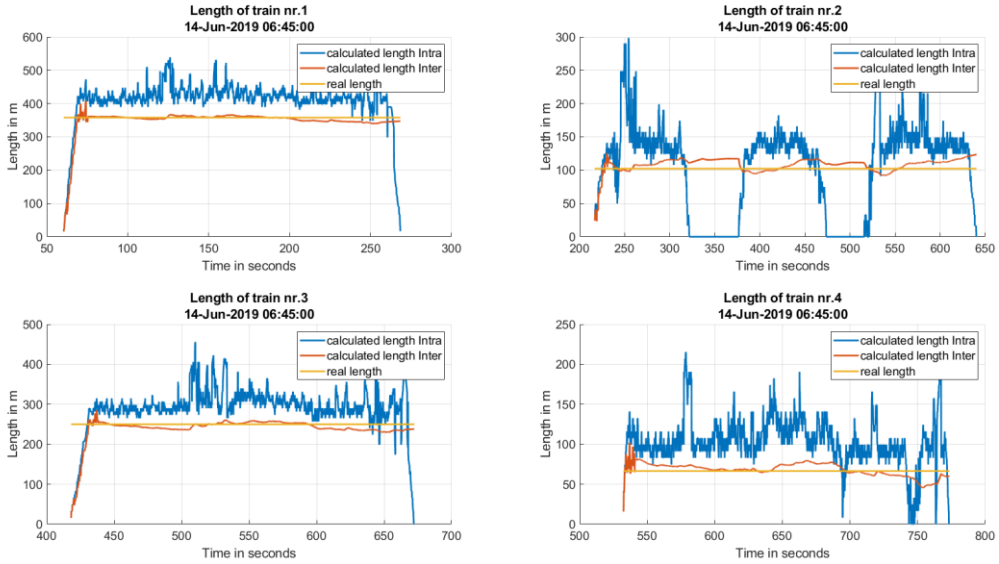

Figure 4-51 The calculated train length compared to the real length. In blue the train length after the Intra Channel Analysis, in red the corrected train length after Inter Channel Analysis and in yellow the real length.

Figure 4-52 shows the corresponding boxplot to Figure 4-51 containing the errors of the corrected length to the real length. The initialization phase at the beginning of the tracking, where the train is not fully present in the measurement area, is excluded in the error calculation. All values are within the interval +-20m.

In the top right plot of Figure 4-51 the blue line shows the train length calculated after the Intra Channel Analysis. The train gets “lost” when the train stops which leads to a calculated length of 0m for the Intra Channel Analysis. Standing trains cannot be measured with FOS because they do not produce vibration in that time. Nevertheless, with the help of the Kalman filter the train can still be followed here by predicting its position. We manually analysed the part when the train gets “lost” to make a statement about the speed at which the train disappears. It turned out that this happens at speeds of around 10m/s. In

section 6.3.4, when comparing results with GNSS even smaller speeds are measured with our algorithm.

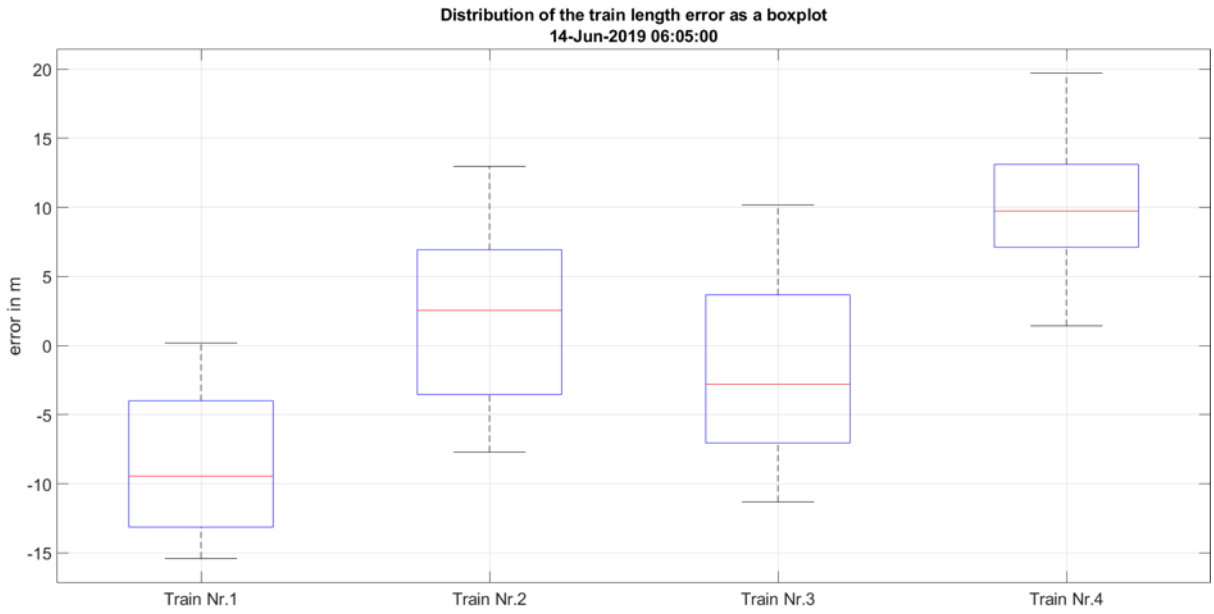

Figure 4-52 Boxplot showing the distribution of the train length error for all four trains. The initialization phase was excluded because there the train is not fully present in the measurement area.

4.5.6 Results

The advanced tracking discussed above was applied on the whole data set from 14 June 2019, which covered 6:55 hours. Since it is known that the FOS measurements are dependent on temperature, the tracking algorithm was also applied to the measurements at low temperatures on 15 December 2019, which covered 2 hours.

All parameters and constants from both the Intra Channel Analysis and the Inter Channel Analysis were calculated using the data of a 1 hour period from 14 June 2019 and are fixed for the evaluation of the rest of the available data.

With the whole data, the algorithm was used in order to test its reliability in finding the trains and its accuracy in calculating their lengths. The real lengths of the trains travelling in these periods were provided by a train schedule. Table 4-2 shows the results regarding the number of trains which were successfully tracked (123 for 14 June 2019) compared to the amount of trains listed in the train schedule of that measurement day. For the data of 14 June 2019 the algorithm found 2 more trains than what was listed in the schedule. A manual review of the measured data showed that these are really existing trains and not errors of the algorithm. For the data measured on 15 December 2019 30 of 30 trains were tracked by the algorithm.

Table 4-2 Availability based on trains that have been successfully tracked.

Availability (Tracked trains)	Target amount -	Actual amount -	Percentage %
14 June 2019	121	123	101.65
15 December 2019	30	30	100

Table 4-3 shows the accuracy of the FOS algorithm based on the error between the calculated and real lengths from the train schedule. The calculated length of each of the 123 trains was compared with the reference length from the train schedule at all times, which leads to a large amount of data points. The table shows that the error of 87.13% of the data points from 14 June 2019 lies within 20m. The appropriate distribution for the data of 14 June 2019 is shown in Figure 4-53 with the calculated gaussian fit in red. A nearly zero mean distribution with a precision of 9.0383m was estimated. The estimated precision is close to the channel length of about 8m.

The distribution for the data of 15 December 2019 is shown in Figure 4-54. The result of the gaussian fit is again nearly zero mean with a precision of 7.218m. The results show very well that the parameters, which were estimated using a short period of the data from 14 June 2019 also achieved very good results on the data from 15 December 2019. A temperature dependence cannot really be detected. Results seem to be even better.

Table 4-3 Accuracy based on the absolute error between calculated length and real length.

Accuracy (length error)	Data points	< 5m %	< 10m %	< 15m %	< 20m %	Min m	Max m
14 June 2019	339907	40.89	63.47	78.03	87.13	$3.45e^{-12}$	84.19
15 December 2019	78975	43.82	67.17	77.73	86.02	$8.57e^{-05}$	62.43

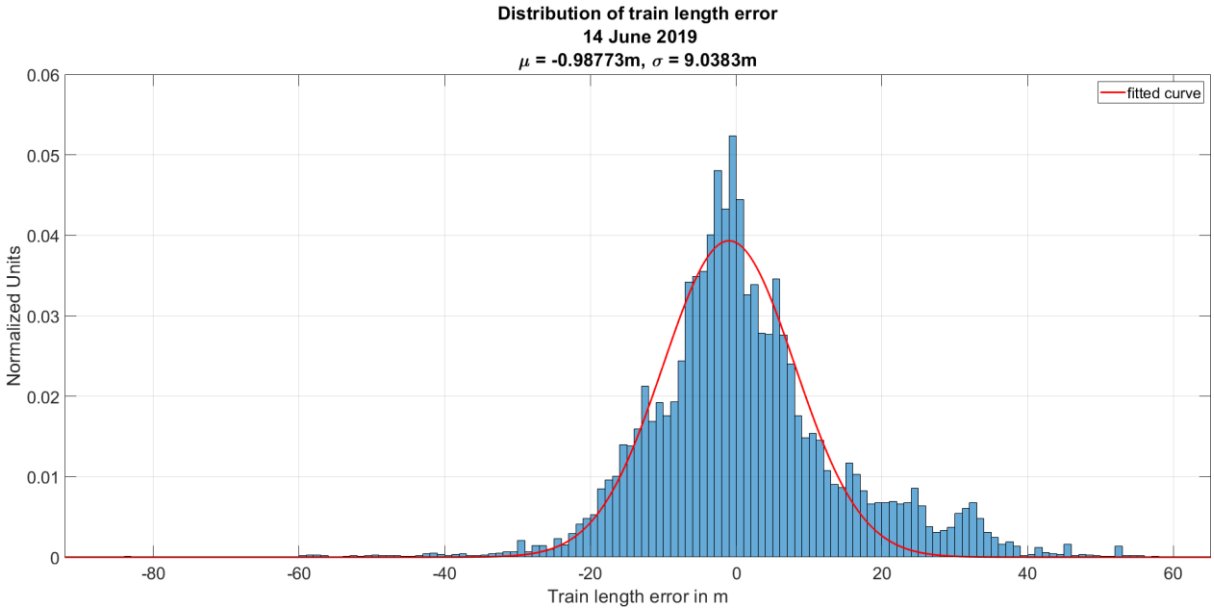

Figure 4-53 Distribution of the train length error for all trains measured on the 14 June 2019. The gaussian fit calculated a nearly zero mean distribution with a precision of 9.0383m.

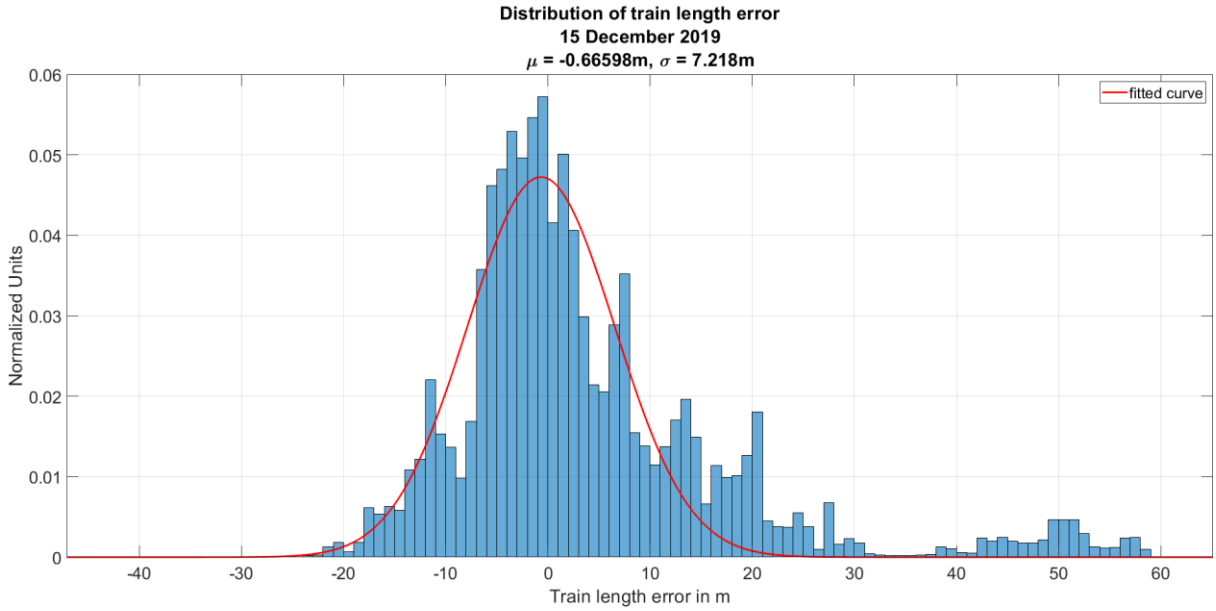

Figure 4-54 Distribution of the train length error for all trains measured on 15 December 2019. The gaussian fit calculated a nearly zero mean distribution with a precision of 7.218m.

Figure 4-55 shows the distribution of the train length error for every tracked train on 14 June 2019. On the x-axis the 123 trains are plotted. The y-axis is the train length error shown as a boxplot. The blue boxes show the range of 50% of the data points and the red marker inside the box is the median error. The dashed lines mark the range of the remaining points except if there are outliers which are marked with red crosses. The measurement train (mewa12) is marked with the dashed green lines and all drives with it show very good results. Figure 4-56 shows the distribution for every tracked train on 15 December 2019.

For most of the trains the error lies within the range of +-20m. There was not enough time for further investigation for the trains where the train length calculation did not fit well. This should be a point to be included for a possible next step. It is not impossible for the schedule to also contain some wrong data regarding the train lengths and, in this case, the data should be compared with a revised schedule.

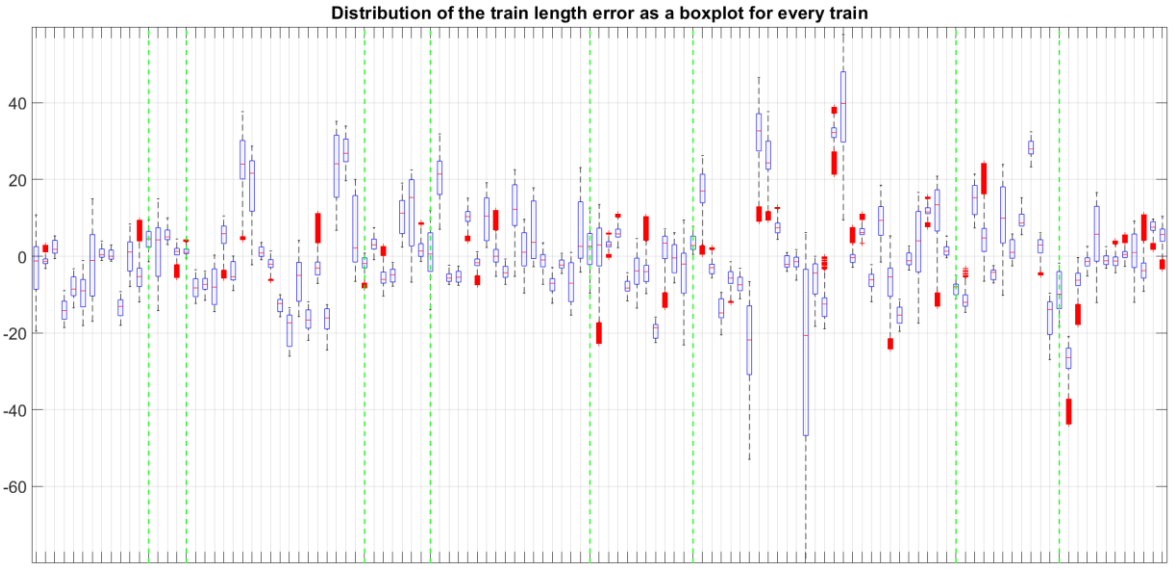

Figure 4-55 Distribution for every tracked train on 14 June 2019 shown as a boxplot. On the x-axis the tracked trains are plotted and on the y-axis the error of true length to calculated length. The measurement train (mewa12) is marked by the dashed green lines.

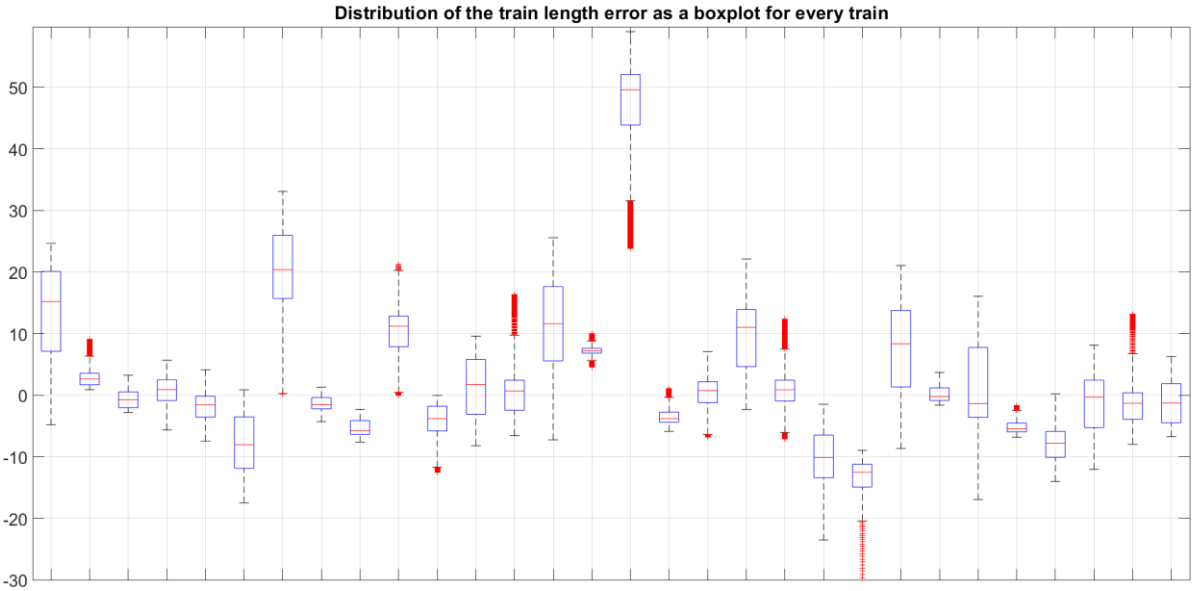

Figure 4-56 Distribution for every tracked train on 15 December 2019 shown as a boxplot. On the x-axis the tracked trains are plotted and on the y-axis the error of true length to calculated length.

5 Multi-sensor Setup

5.1 Introduction

Every sensor has its strengths and weaknesses and is available or unavailable under certain conditions (e.g. GNSS is not available in tunnels). Hence, no sensor can fulfil the required SIL (e.g. SIL 4) as a standalone system for highly available, accurate and safe localisation over all use cases. A possible solution to this problem is to combine two or more independent and diverse sensors to sensor systems (see Figure 5-1). In order to achieve the desired SIL, a suitable combination of sensor signals is needed. For systematic investigation on identifying these suitable combinations, all possible sensors and their characteristics are first documented in a morphological box (cf. [1]). In a second step the independent sensors with different types of errors can be combined to sensor systems.

Regarding the ongoing Proof of Concept (PoC) the following sensor systems are assessed and compared:

- GNSS / IMU / Wheel odometry
- Visual Odometry /Video landmarks
- Balises / Wheel odometry
- FOS

While the first two sensor systems are new inventions, the third one is state of the art and it is used in ETCS. Another promising sensor system which was not considered in this PoC might be the combination of Video / IMU.

Choosing an x-out-of-y system architecture (y >=x) increases the safety of the localisation. In this architecture y different independent sensor systems are taken into account while at least x sensor systems have to deliver similar results to gain a valid sensor value. Otherwise the sensor signals for this time stamp are not valid. If x or more sensor system signals are similar, the sensor data fusion chooses the most appropriate sensor system data for localisation according to a predefined criterion, e.g. the confidence level. Figure 5-1 shows an example of a x-out-of-y system architecture.

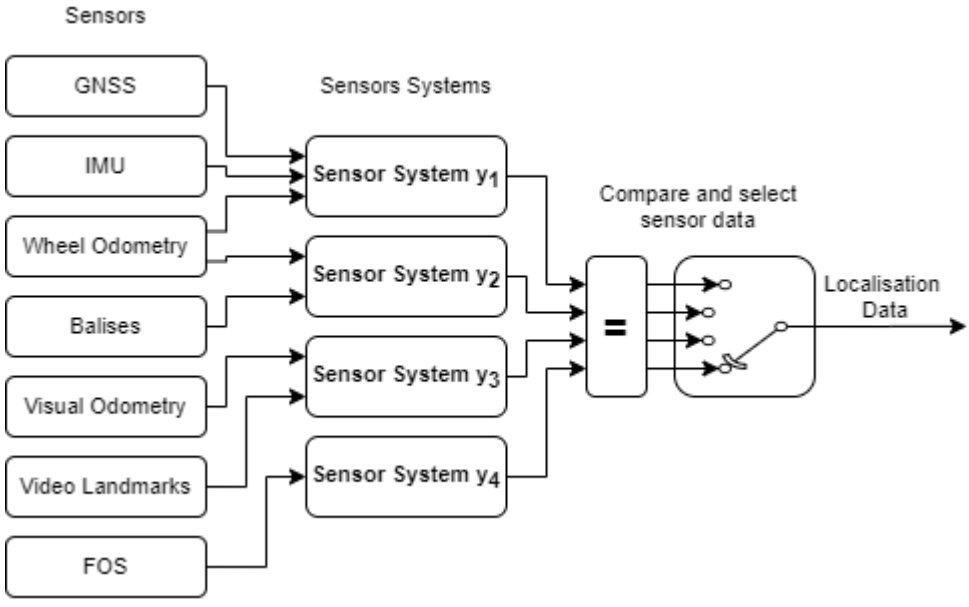

Figure 5-1: Example of a x-out-of-y system architecture

5.2 Consideration of certification

Considering the certification of sensor systems in a x-out-of-y system architecture, the aim should be to separate the sensor signal processing and the fusion algorithm from a monitoring function in a risk-based approach. Figure 5-2 [26] shows an example of such a separated architecture. In this case, the monitoring function requires high SIL (e.g. SIL 4) and it is framed by a grey box.

The major benefit of this approach is that only the monitoring and voting functions have to developed according to the CENELEC standard with the given SIL. The sensors have only to be qualified to be used in railway environment. Cross acceptance from other domains like aerospace would also be possible. A certificate for qualification of sensors can be obtained by independent accredited test laboratories according to Figure 5-3.

The development of such a system can be faster, more economical and lower in risk and the certification is easier to achieve. Only the monitoring function needs to fulfill an appropriate SIL and a CENELEC compliant development process.

More details about certification can be found in [27].

What is more, even a sensor system using machine learning approaches could be deployed in such an architecture, if qualification is possible.

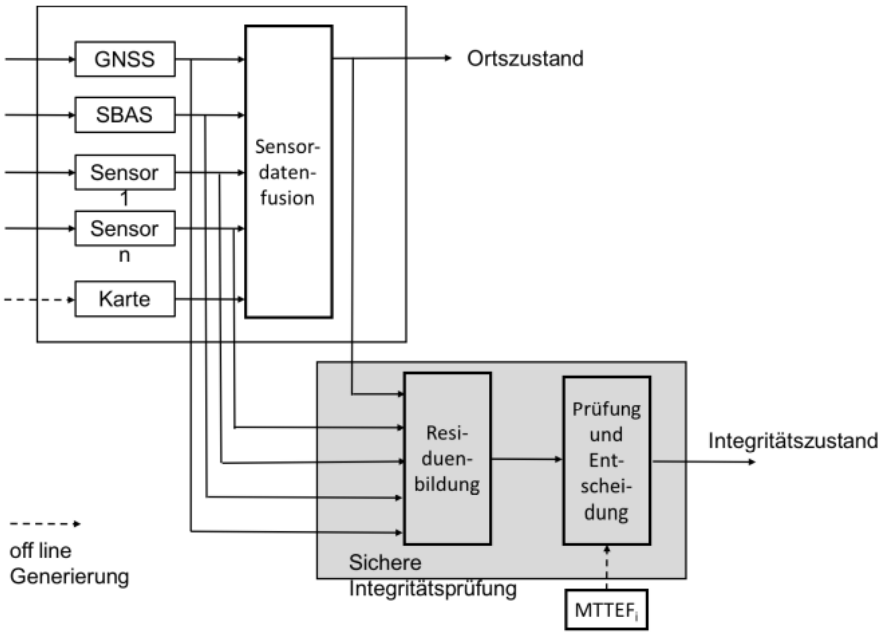

Figure 5-2: Functional architecture of a localisation system with safe integrity check (from "Machbarkeitsstudie für eine genaue, sichere Lokalisierung" [26])

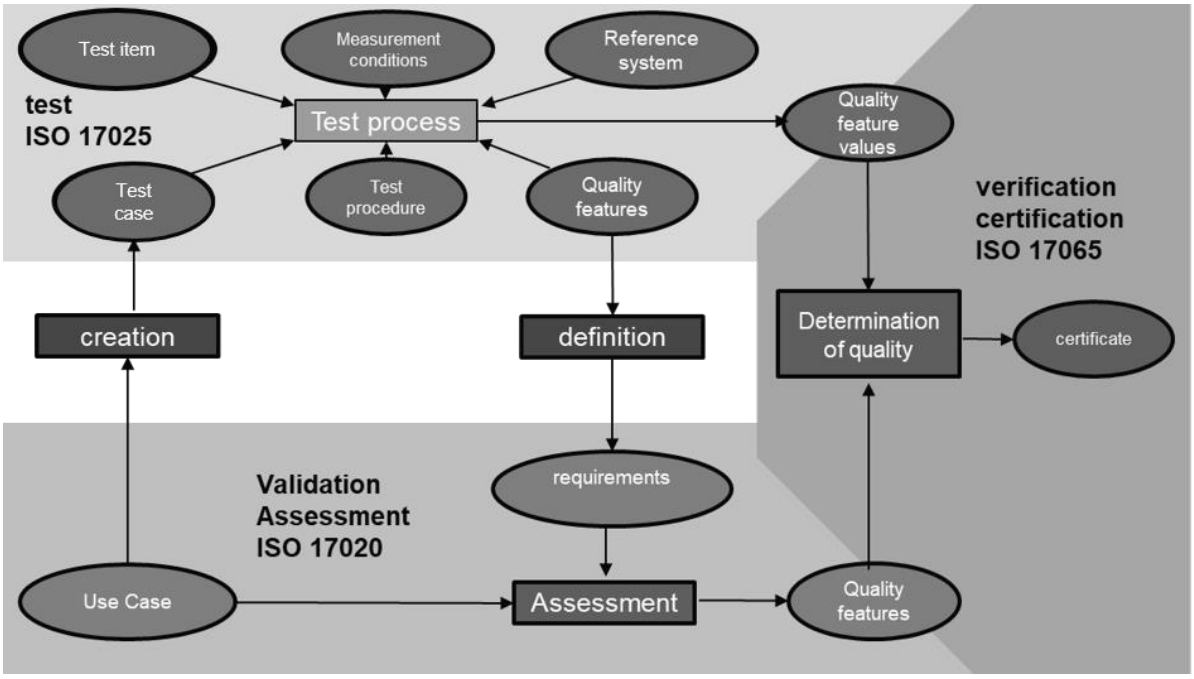

Figure 5-3: Overview: Qualification of a sensor system [27])

6 Measurement runs

6.1 Overview of measurement runs

The following evaluations were carried out in the context of this report:

Date	Route	Type	Sensors
14.06.19	Ostermundigen - Thun	Measurement with SBB diagnostic vehicle	Video; FOS; GNSS/IMU;
23.10.19	Münsigen - Uttingen	Calibration FOS System	FOS only
03.12.19	Bern	Measurement in Depot	Video only
15.12.19	Münsigen - Uttingen	FOS Data extraction (low temparature)	FOS only
21.01.20	Münsigen - Uttingen	FOS Data extraction (temparature below freezing)	FOS only
05.02.20	Bern - St. Gallen - Bern	Measurement run with Regular train	Video only
12.02.20	Bern - Brig - Bern	Measurement run with Regular train	Video only
04.03.20	Biel - Lausanne - Biel	Measurement run with tilting train (ICN)	Video only

Table 6-1 Overview of measurement runs

6.2 Ground truth

In the following the background for assessing the various technologies and comparing them to a valid and accurate ground truth is described according to [28].

“In SIL 4 approved ETCS Level 2 operation, a train sends its train position report (TPR) once every 6 seconds (SBB configuration), on average, to the Radio Block Center (RBC). The TPR consists of the last passed balise (group) and the travelled distance from there as well as the direction of travel. To overcome the issues of certification, in a 1st stage it is proposed to prove that the new localisation system has the same performance regarding quality and safety than the certified one (GAMAB principle). Therefore, with the new localisation system (regardless of the technology) the same TPRs have to be generated with at least the same quality. The successful comparison of the statistical relevant number of TPRs between the current and new systems can be used to get approval according to e.g. CSM 2013/402/EC.

In addition to the balises, axle counters could also be used, since they are also approved for SIL 4.

Within video localisation with GNSS synchronized time, global drift compensation can be done with artificial reference points (e.g. AprilTags) and rail infrastructure other than balises (e.g. points, bridges, catenary masts). However visual balise or axle counter detection will be used - without the need for a separate sensor – in order to trigger the TPR generation or time synchronization for comparison purposes. With a successful proof of the same or better quality of the new system, ”artificial balises” can be introduced wherever required to meet the requirements for realizing moving block.

At the same time, this will lead to a lean and promising migration strategy (no change of the ETCS interface to the RBC).”

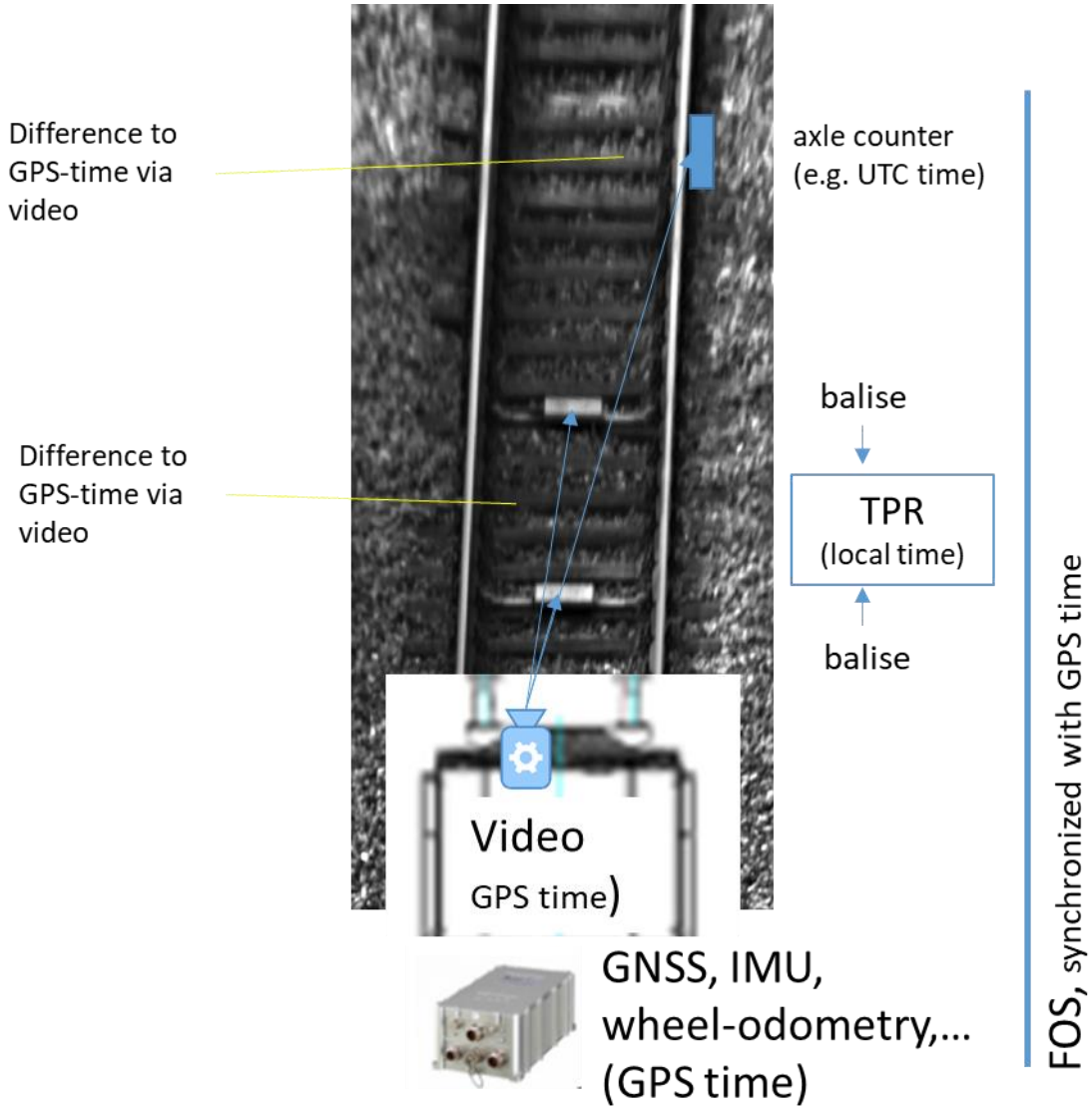

Figure 6-1 - time synchronization and references for ground truth

Time stamps of axle counter log-files and TPRs can be adjusted to GPS time when detected in the video frame. Distance from camera to axle counter or balises resp. can be calculated according to [2]. With known speed, the time when passing the balises or axle counters can be derived. Time resolution depends on the frame rate and is equal to 16.7 ms when using a 60 Hz frame rate, which is our case.

For the comparison (and fusion) of data of different sensors it is important to compensate for the different latencies. An event at a given time t will be available on the hardware output of the sensor after an acquisition time t_{ac} (e.g. reading out the CCD chip or analogue to digital conversion). The digital processing (e.g. image or signal processing, filtering) will need another time t_{pr} to be completed. That means that the sensor system output for the event at time t will be ready to use at time $t+t_{ac}+t_{pr}$. Assuming real-time processing the worst-case latency will be twice the sampling time. If there is any phase lag e.g. due to windowing/filtering it can be even more. For each sensor system the latencies have to be determined or estimated.

With all sensors synchronized to the same time source, taking into account the different sensor dependent latencies, it is possible to compare them all at the given reference positions of the axle counters and

balises. For the measurements on June 14th, we had access to log files of 5 axle counters which were used to compare all sensor technologies involved, which are detailed in the next section.

According to current discussions with certification experts, this approach is suitable to be accepted as standard operational procedure for the qualification of sensors.

6.3 Results and Comparison

6.3.1 Visual Odometry (1D)

6.3.1.1 Traveled distance: comparison with GNSS / IMU and GTG

The travelled distance calculated with Visual Odometry is compared to a combination of GNSS / IMU and to track topography (GTG). The measured distance with Visual Odometry is in accordance, within the given systematic uncertainty, with the distance measured with GTG and GNSS/IMU (Figure 6-2).

By comparing the position calculated with Visual Odometry with GTG and GNSS/IMU measurements, the estimated systematic uncertainties (listed in Table 6-2) seem to be conservative. A measurement with lower systematic uncertainty is out of the scope of this report. It shall be noted from Table 3-5, that the main source of systematic uncertainty is due to the estimation of the camera extrinsic parameters and the determination of the absolute scale using the railway track width as reference. The automatic procedure for the estimation of those parameters allows for an easy implementation of the camera system for data collection. However, in this case, the uncertainty is higher with respect to a fixed camera whose initial pose could be estimated with a calibration sheet (chessboard).

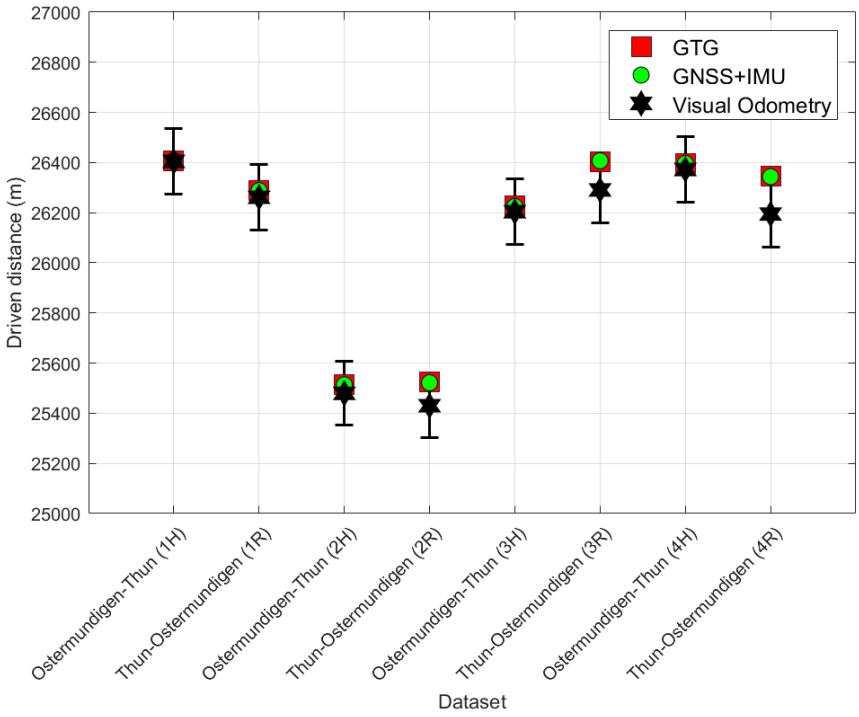

Figure 6-2: The driven distance is calculated with systematic uncertainty (black) and compared to the GNSS/IMU (green) and GTG (red) measurements.

Table 6-2: The calculated distance values with systematic uncertainty are listed and compared with GNSS/IMU and GTG. * Due to problem to the dataset with GNSS/IMU, data containing GNSS only are used.

Drive	GTG	GNSS/IMU	Visual Odometry			
	Value (m)	Value (m)	Value (m)	Uncertainty (m)	Relative difference to GTG (%)	Relative difference to GNSS (%)
OT_1H	26406	26408	26405	202	0.0	0.0
OT_1R	26297	26288	26261	180	-0.1	-0.1
OT_2H	25515	25513	25479	197	-0.1	-0.1
OT_2R	25524	25522	25430	173	-0.4	-0.4
OT_3H	26230	26225	26203	203	-0.1	-0.1
OT_3R	26403	26407*	26291	181	-0.4	-0.4
OT_4H	26402	26397	26373	204	-0.1	-0.1
OT_4R	26352	26344	26195	180	-0.6	-0.6

6.3.1.2 Traveled distance: comparison with balises

The calculated distance is also compared to the distance between balises.

The balises can be identified in the collected images. The first balise of each balise group is taken for the calculation of the distances. Figure 6-3 shows the nominal distance (black) from DfA, the distance calculated by using the GNSS/IMU combined measurement (red) and the distance calculated by Visual Odometry (blue). Figure 6-4 shows the relative distance between the nominal values and the values measured by Visual Odometry (blue) and by the GNSS/IMU combination (red). As it can be seen, a high grade of congruency can be achieved by comparing the nominal distance with the one calculated with Visual Odometry. In Table 6-3, the calculated distance between consecutive balises is summarized.

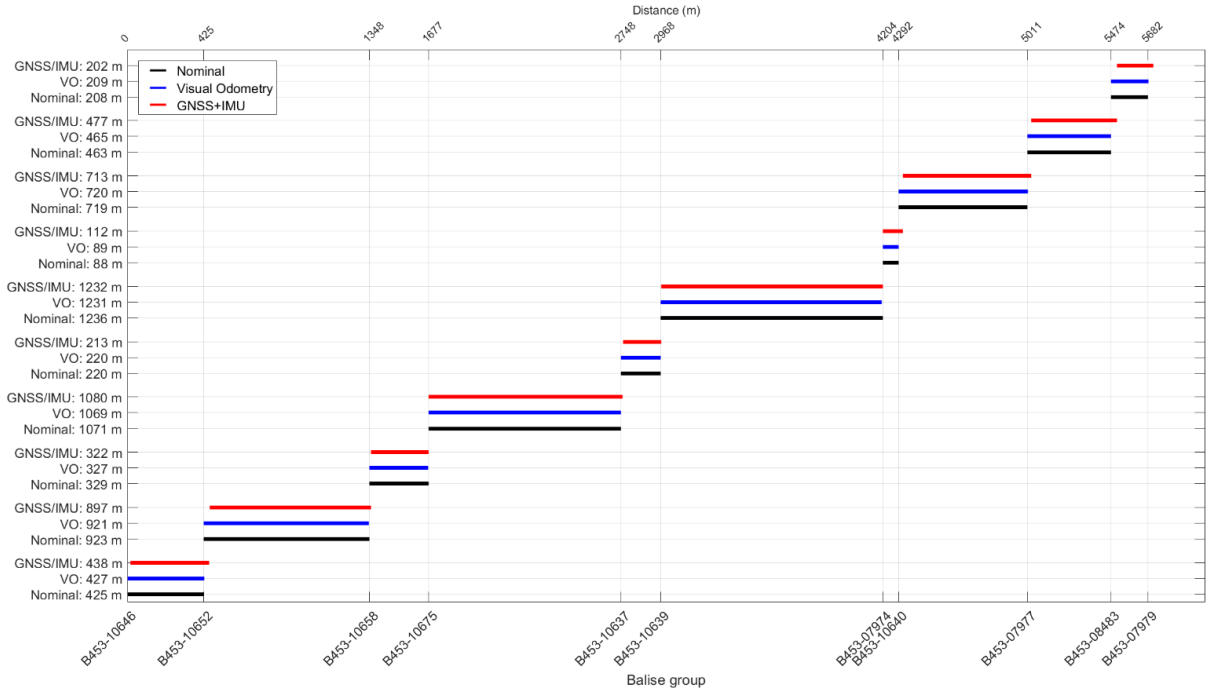

Figure 6-3: The nominal distance (black) from DfA, the distance calculated by using the GNSS/IMU combined measurement (red) and the distance calculated by Visual Odometry (blue) are compared.

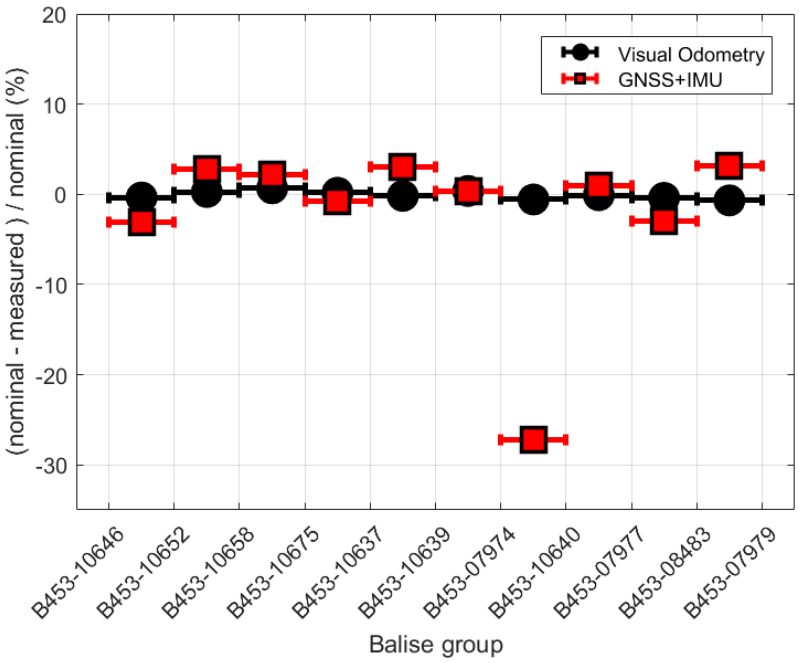

Figure 6-4 The distance between nominal and measured values, divided by the nominal distance is shown. The measured values are from Visual Odometry (red) and from GNSS/IMU (black).

Table 6-3: The results of the measured distance between two consecutive balises (distance between column *Balise A* and column *Balise B*) are summarized. Data of balises are taken from DfA.

Balise A	Balise B	Nominal distance (m)	VO distance (m)	VO difference (%)	GNSS+IMU distance (m)	GNSS+IMU difference (%)
B453-10646	B453-10652	425	426.5	-0.4	438.3	-3.1
B453-10652	B453-10658	923	920.6	0.3	897.4	2.8
B453-10658	B453-10675	329	326.8	0.7	321.8	2.2
B453-10675	B453-10637	1071	1069.2	0.2	1079.7	-0.8
B453-10637	B453-10639	220	220.4	-0.2	213.4	3.0
B453-10639	B453-07974	1236	1231.4	0.4	1232.1	0.3
B453-07974	B453-10640	88	88.5	-0.6	11.9	-27.2
B453-10640	B453-07977	719	719.7	-0.1	712.6	0.9
B453-07977	B453-08483	463	464.8	-0.4	477.0	-3.0
B453-08483	B453-07979	208	209.4	-0.7	201.6	3.1

6.3.1.3 Speed: comparison with GNSS / IMU

The absolute speed of the train is calculated from the traveled distance and the framerate of the camera. In Figure 6-5 (left) to Figure 6-12 (left), the absolute speed calculated with Visual Odometry (blue) is compared with the one measured by the combination (red) of GNSS and IMU for drives OT_1H to OT_4R. It can be seen, that the shape of the measured speeds by Visual Odometry matches the shape of the speeds measured by GNSS combined with IMU, in all the data runs analysed.

The distribution of the difference between the speed value measured from the combination between GNSS and IMU with the one measured with Visual Odometry, is shown in Figure 6-5 (right) to Figure 6-12 (right). From a gaussian fit of the distribution, the mean value of the speed difference and the standard deviation are estimated.

Table 6-4 shows the fit results. The trueness is defined as the mean value of the difference between the speed measured by the reference (GNSS combined with IMU) and the speed measured by Visual Odometry. The precision is the standard deviation of the distribution of the speed difference or in other

words: the precision of the localisation is within the accuracy of the ground truth. As it can be observed, larger values of the trueness and larger values of the precision are observed in drives where the locomotive drives backwards (drives OT_1R, OT_2R, OT_3R, OT_4R) with respect to drives where the locomotive drives forwards (drives OT_1H, OT_2H, OT_3H, OT_4H). This forward/backwards asymmetry is under investigation and is observed also in Table 6-2, where the relative difference of the traveled distance with respect to GNSS/IMU is slighlty larger when the locomotive drives backwards.

Table 6-4 Trueness and precision of the difference of the speed measured with GNSS combined with IMU and Visual Odometry. * Due to problem to the dataset with GNSS/IMU, data containing GNSS only are used.

Drive	Trueness (km/h)	Precision (km/h)
OT_1H	0.06	0.76
OT_2H	0.12	0.57
OT_3H	0.05	0.98
OT_4H	0.09	0.76
OT_1R	0.17	0.90
OT_2R	0.24	0.78
OT_3R*	0.37	0.80
OT_4R	0.65	0.79

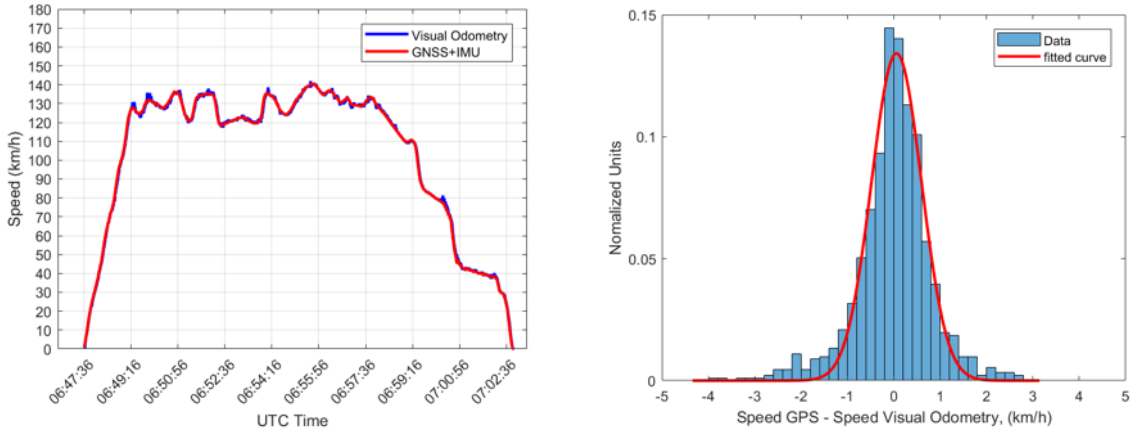

Figure 6-5 (left) The absolute speed measured with Visual Odometry (blue) is compared to the one measured with the combination of GNSS and IMU (red) during the drive OT_1H. (right) The distribution of the speed difference between the one measured with GNSS combined with IMU and the one calculated with Visual Odometry is shown for drive OT_1H.

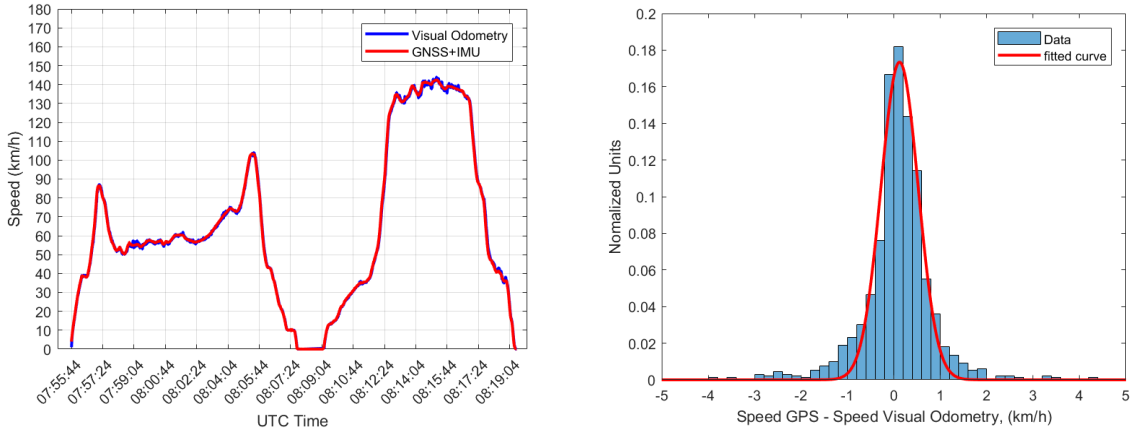

Figure 6-6 (left) The absolute speed measured with Visual Odometry (blue) is compared to the one measured with the combination of GNSS and IMU (red) during the drive OT_2H. (right) The distribution of the speed difference between the one measured with GNSS combined with IMU and the one calculated with Visual Odometry is shown for drive OT_2H.

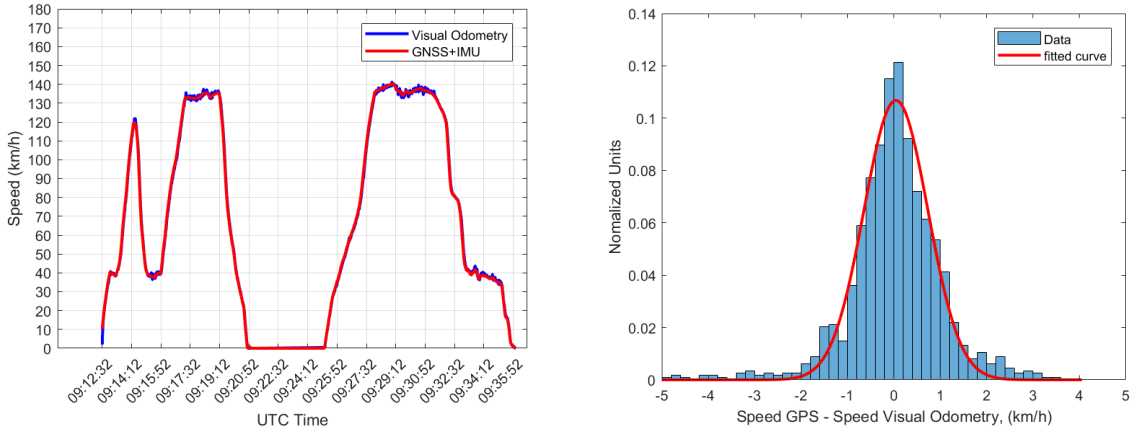

Figure 6-7 (left) The absolute speed measured with Visual Odometry (blue) is compared to the one measured with the combination of GNSS and IMU (red) during the drive OT_3H. (right) The distribution of the speed difference between the one measured with GNSS combined with IMU and the one calculated with Visual Odometry is shown for drive OT_3H.

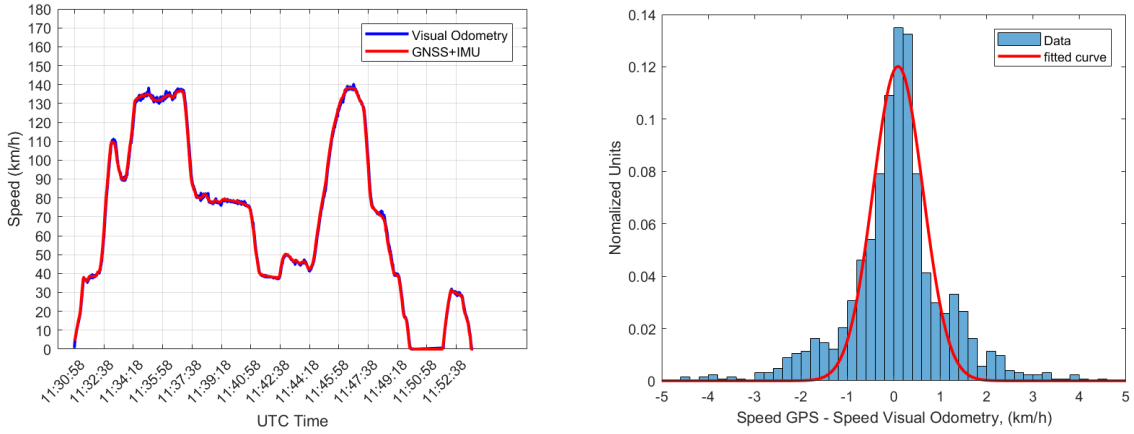

Figure 6-8 (left) The absolute speed measured with Visual Odometry (blue) is compared to the one measured with the combination of GNSS and IMU (red) during the drive OT_4H. (right) The distribution of the speed difference between the one measured with GNSS combined with IMU and the one calculated with Visual Odometry is shown for drive OT_4H.

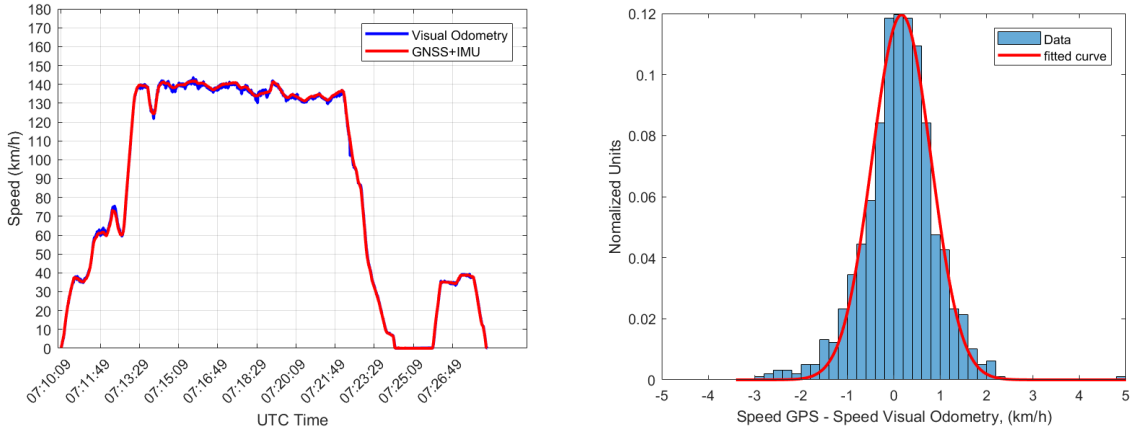

Figure 6-9 (left) The absolute speed measured with Visual Odometry (blue) is compared to the one measured with the combination of GNSS and IMU (red) during the drive OT_1R. (right) The distribution of the speed difference between the one measured with GNSS combined with IMU and the one calculated with Visual Odometry is shown for drive OT_1R.

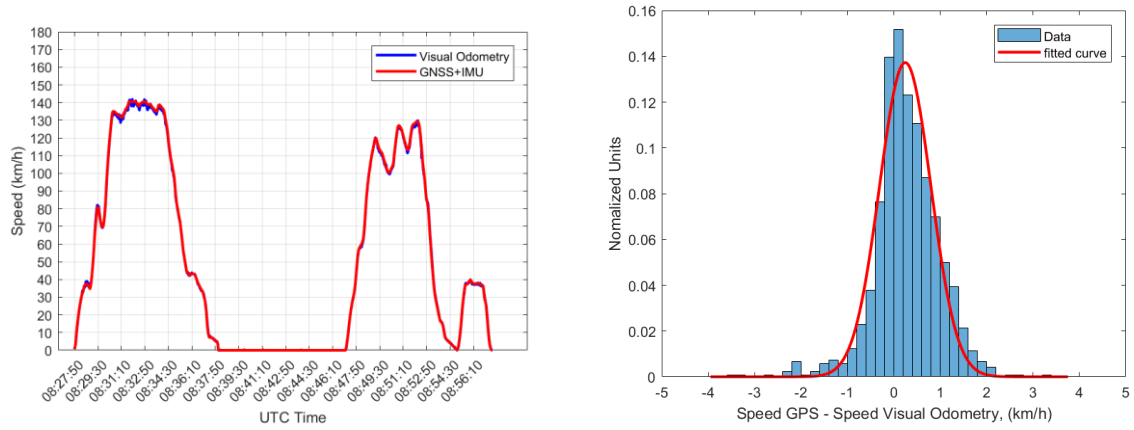

Figure 6-10 (left) The absolute speed measured with Visual Odometry (blue) is compared to the one measured with the combination of GNSS and IMU (red) during the drive OT_2R. (right) The distribution of the speed difference between the one measured with GNSS combined with IMU and the one calculated with Visual Odometry is shown for drive OT_2R.

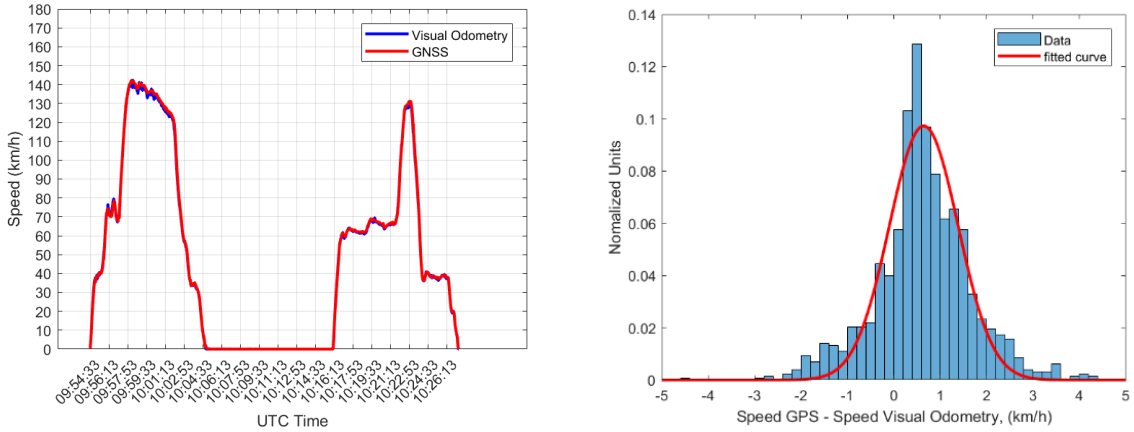

Figure 6-11 (left) The absolute speed measured with Visual Odometry (blue) is compared to the one measured with GNSS (red) during the drive OT_3R. (right) The distribution of the speed difference between the one measured with GNSS combined with IMU and the one calculated with Visual Odometry is shown for drive OT_3R.

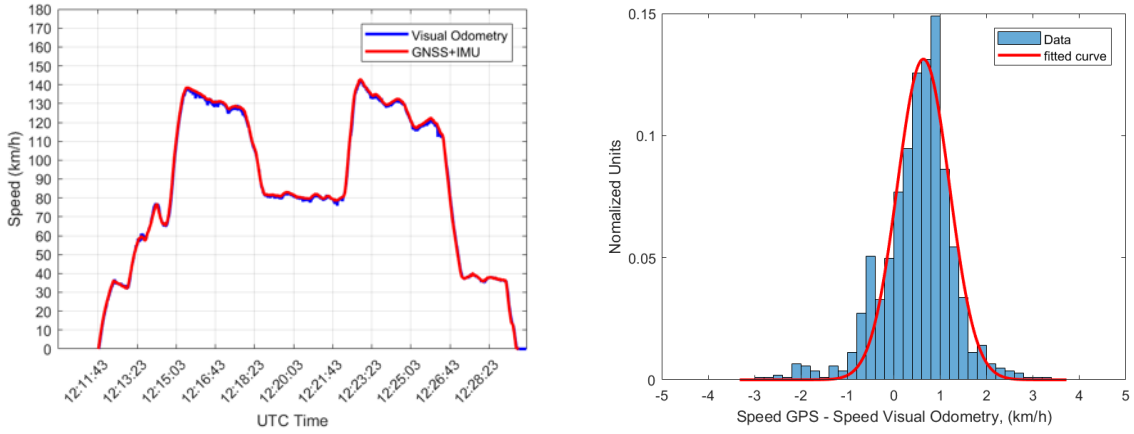

Figure 6-12 (left) The absolute speed measured with Visual Odometry (blue) is compared to the one measured with the combination of GNSS and IMU (red) during the drive OT_4R. (right) The distribution of the speed difference between the one measured with GNSS combined with IMU and the one calculated with Visual Odometry is shown for drive OT_4R.

6.3.2 Visual Odometry (3D)

Figure 6-13 shows the calculated path (in cyan) for each of the four measurements taken on the track from Ostermundingen to Thun, using data collected by the front camera. The calculated path is compared to the combined measurement from GNSS and IMU (in red). Since the GTG data are very close to the GNSS/IMU measurement, the former are not displayed.

Measured data with Visual Odometry and GNSS/IMU combination match at the beginning of any path. A drift is expected as the traveled distance increases and it can be observed more or less in any of the runs.

It shall be noted, that the measurements from Visual Odometry presented here do not rely on any absolute reference. This means that it could be the case that wrong measurements of few image frames cannot be corrected and errors in the calculated train rotation propagate till the end of the drive. This is probably the reason for the clearly visible difference between the GNSS/IMU reference and the measured data observed in path OT_2H, where measurement are in accordance up to half of the path, and then the position calculated from the Visual Odometry drifts away.

The results highlight that visual odometry can be very precise on a short scale (~few kilometers) but needs an absolute reference to be precise on larger scale.

If the detection of AprilTags, points or catenary masts is added to the video odometry the position can be corrected.

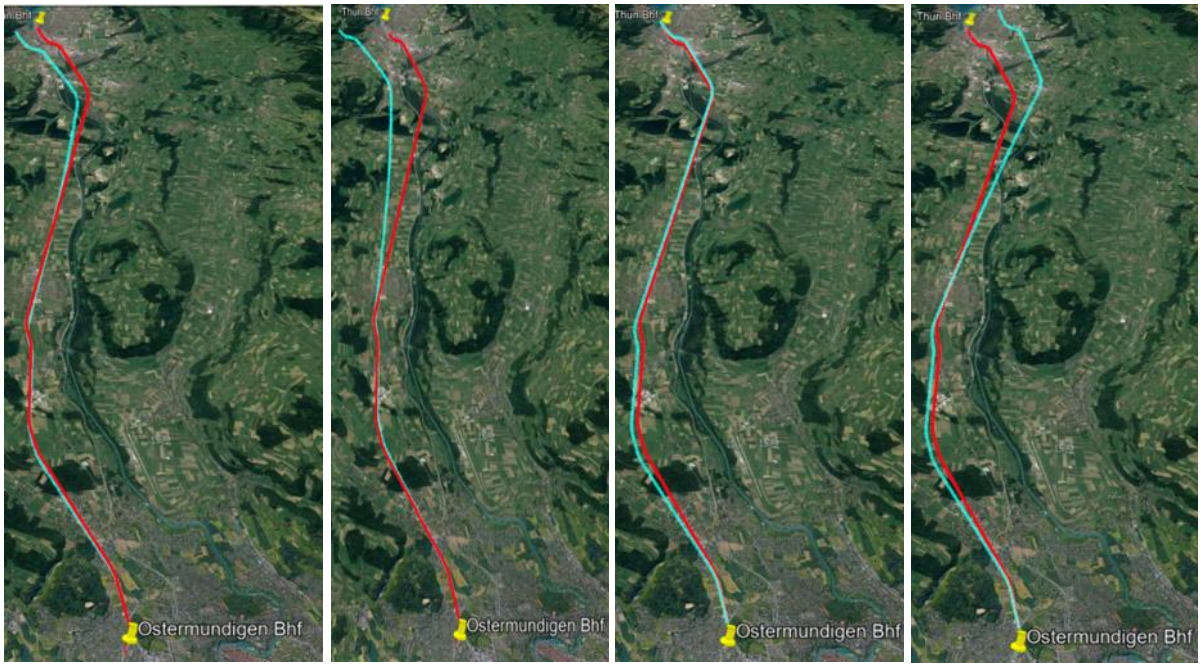

Figure 6-13: The calculated position (cyan) for the 4 drives (from left to right: OT_1H, OT_2H, OT_3H, OT_4H) from Ostermundigen to Thun and compared to the GNSS/IMU reference (red)

6.3.3 Video Localisation

The train position, calculated with Visual Odometry, can be more precise by using global references. In the following, the fixed positions of the railway points are used as reference.

Table 3-4 shows that railway points were detected during the drive OT_1H. Once the frog of a railway point is identified in the image, the calculated local position is referenced to that value. This allows for a reset in the accumulated drift distance and a correction of the train direction.

Figure 6-14 shows the comparison between the train position calculated with Visual Odometry only (cyan) and the position calculated with Visual Odometry by using the railway points as reference (yellow). The railway points are shown (green triangles) in Figure 6-14. The improvement can be seen by comparing both positions with the GNSS / IMU combination. A drift of the position calculated by Visual Odometry is visible and can be strongly reduced by the introduction of the global reference. The calculated train position is very much in accordance with the GNSS / IMU combination, although small deviations are observed at the end of the drive, where the train approaches the station in Thun.

Figure 6-14: The position calculated with the Visual Odometry (cyan) is corrected by using global reference like railway points (green triangles). The corrected position (yellow) is compared with the GNSS / IMU combination (red).

The distribution of the difference between the position measured from the GNSS/IMU combination and the one calculated with Visual Odometry by using railway points as reference, is shown in Figure 6-15.

The distribution peaks at 35 meters. This is the distance from the front of the train (where the camera is located) to the IMU. A gaussian fit has been performed and the results are shown in Table 6-5.

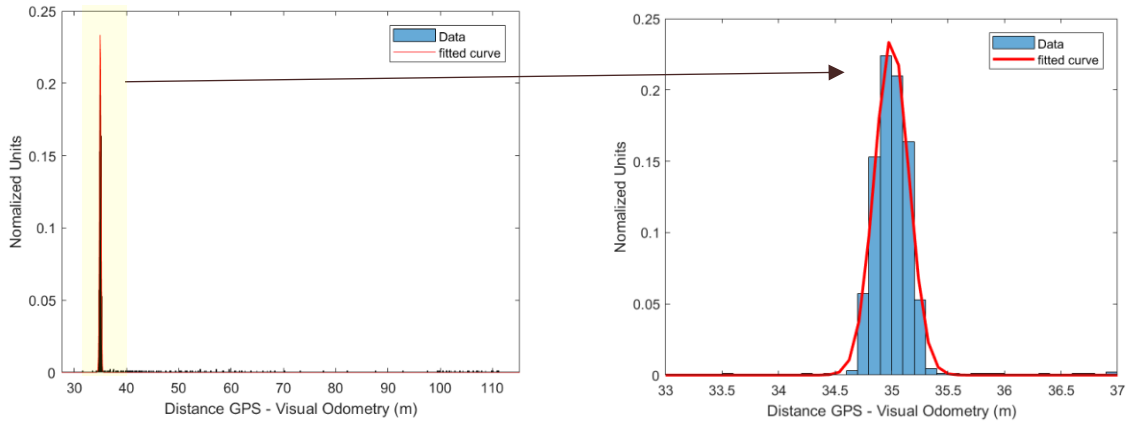

Figure 6-15 The distribution of the difference between the position measured from GNSS/IMU combination and the one calculated with Visual Odometry by using railway point as reference. The distribution peaks at 35 meters. This is the position of the IMU, that is located 35 meters back with respect to the camera system, that is located at the front of the train. (Left) The distribution ranges from 20 to 150 meters and the tails can be seen. (Right) The distribution ranges from 33 to 37 meters.

Table 6-5 Results of the gaussian fit of the distribution of the difference between the position calculated with GNSS - IMU combination and the position calculated with Visual Odometry. The gaussian fit has been performed over the whole distance range, from 20 to 150 meters. The mean value is the estimated distance of the camera (located on the wind-screen) to the IMU. The precision is the width of the distribution.

Drive	Distance to the GNSS antenna (m)	Precision (m)
OT_1H	35.0	0.21
OT_2H	35.8	0.23
OT_3H	35.6	0.21
OT_4H	35.3	0.26

Figure 6-16 shows the comparison between the train position calculated by Visual Odometry and the position from track topography (GTG). The calculated train position is in accordance with GTG, although small deviations are observed at the end of the drive, where the train approaches the station in Thun.

Figure 6-16 The position calculated with the Visual Odometry (yellow) is corrected by using global reference like railway points and compared to GTG (red).

The distribution of the difference between the position measured from the GTG and the one calculated with Visual Odometry by using railway point as reference, is shown in Figure 6-17. The long tails of the distribution are likely due to the irregular spacing of the points of the position measured by GTG. Indeed, large distance between the reference points can occur and could lead to large deviation when compared to the position measured by Visual Odometry. A gaussian fit of the distribution is not suitable due to the long tails.

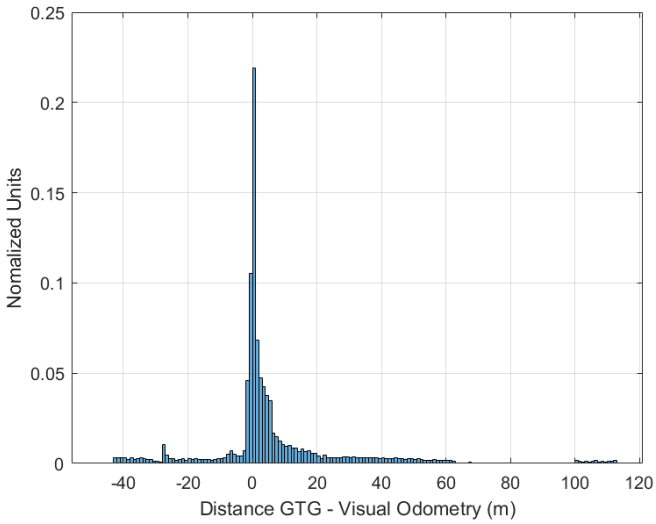

Figure 6-17 The distribution of the difference between the position measured from the GTG and the one calculated with Visual Odometry.

6.3.4 FOS

A GNSS in combination with an IMU as a reference (hereinafter referred to as GNSS / IMU) was provided for the drives of the measurement train on 14 June 2019. This should serve as a reference for the following evaluation. There are two important considerations that have to be made when comparing the two sensors: their time has to be synchronised and the reference point of the GNSS / IMU on the train in relation to either the front or rear end of the train must be known.

It should also be mentioned that the comparison between FOS and GNSS / IMU was done using their lowest common denominator. This means that FOS can supply other values that other sensors cannot, e.g., train length, instantly.

The position of the GNSS / IMU on the measurement train is known and using the mapping between FOS channel and real world coordinates this position can be calculated. The problem with the time

synchronisation was solved by estimating the time delay using the first drive with the measurement train (OT_1H). Figure 6-18 shows the distribution of the time delay during the first drive.

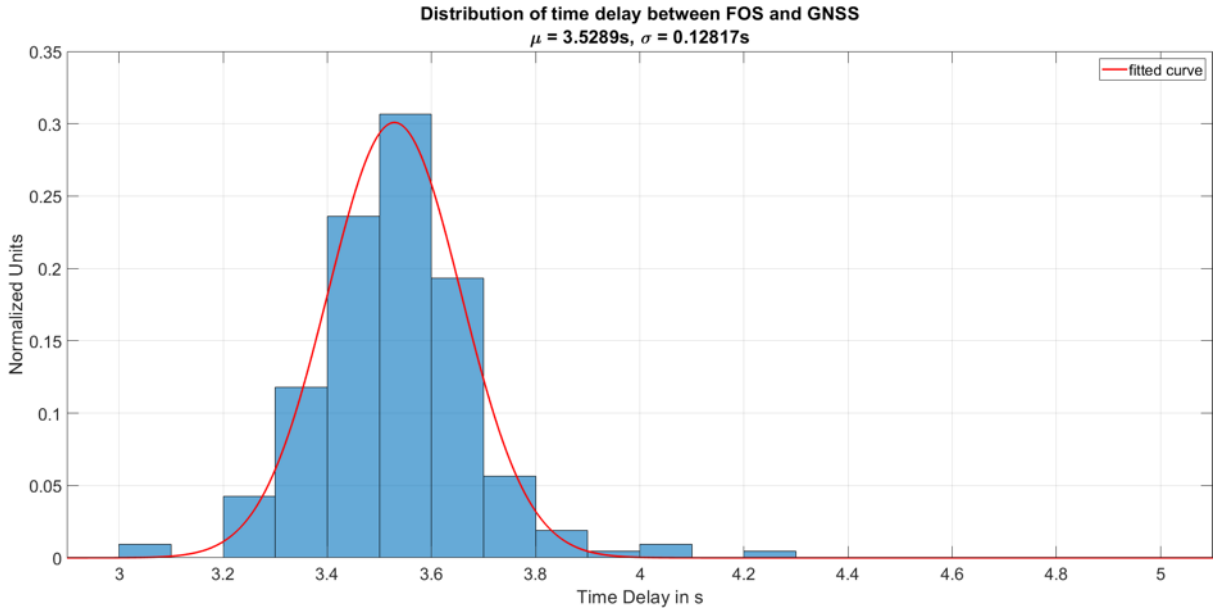

Figure 6-18 Distribution of the time delay between FOS and GNSS / IMU clock during the drive OT_1H.

The FOS time was adjusted by the estimated mean delay of 3.53s for the remaining drives. Figure 6-19 and Figure 6-20 show the localisation errors when comparing GNSS / IMU for drives with the locomotive in front or the control wagon in front, respectively.

The reference position of the antenna was used for the calculation of the time delay using a single measurement drive. This position, however, was not used in the comparisons shown in Figure 6-19 and Figure 6-20.

The measured front position by FOS was compared to the GNSS / IMU position. Therefore, the estimated mean value of the gaussian fit in Figure 6-19 is an estimation of the reference position of GNSS / IMU on the train when the locomotive is in the front position. Figure 6-20 shows the distribution of the localisation error when driving in the other direction and so the mean value of the gaussian fit is an estimation of the reference position of GNSS / IMU on the train when the control wagon is in the front position. The values are compared in Table 6-6. For both directions the reference position of GNSS /

IMU was estimated with a small error if you consider the resolution of about 8m for FOS. The precision, which is the width of the gaussian fit, also lies within the resolution of FOS.

For the measurement train, the sum of the estimated antenna positions should result in the train length which is about 64.52m in this case. The real length of the measurement train is 66.5m.

There are also some other parameters on which the results could depend:

- The estimation of the time delay. Minimum value was 3.07s and maximum value was 4.27s.
- The accuracy of GNSS / IMU as the reference.

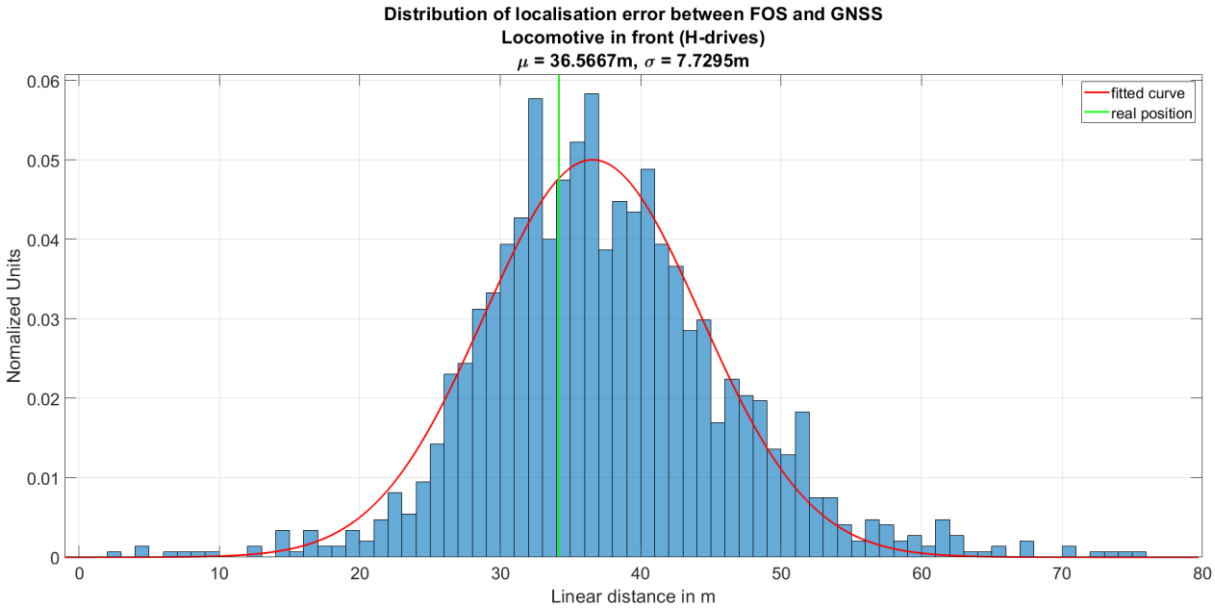

Figure 6-19 Distribution of the localisation error between GNSS / IMU and FOS for all drives with the locomotive in front position using the front position estimated by FOS. The mean value of the gaussian fit is an estimation of the GNSS / IMU antenna position. Their position is known to be 34.18m behind the front of the locomotive.

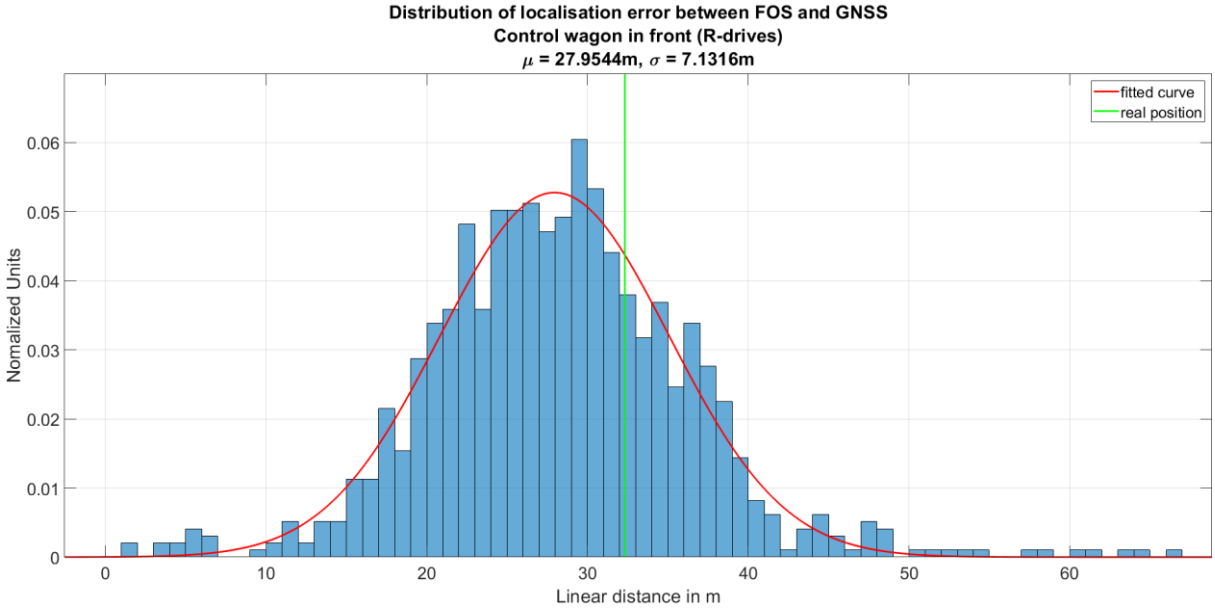

Figure 6-20 Distribution of the localisation error between GNSS / IMU and FOS for all drives with the control wagon in front position using the front position estimated by FOS. The mean value of the gaussian fit is an estimation of the GNSS /IMU antenna position. Their position is known to be 32.32m behind the front of the control wagon.

Table 6-6 Results of the gaussian fit of the distribution of the difference between the position calculated with GNSS / IMU and the position calculated with FOS. The mean value is the estimated distance of the front of the train to the GNSS / IMU reference position on the train. The precision is the width of the distribution.

Localistion error	Real reference position for GNSS	Estimated reference position	Precision of estimation
OT_H (Locomotive in front)	34.18m	36.57m	7.73m
OT_R (Control wagon in front)	32.32m	27.95m	7.13m
Train length	66.5m	64.52m	-

In Figure 6-21 the calculated speed of both sensors is compared for the drive OT_2H. The speed calculated with FOS shows a little more variation, which is due to the resolution of FOS.

The train could be tracked and measured down to a speed of 7 m/s (25km/h).

The distribution of the speed error between FOS and GNSS/ IMU for drives with the locomotive in front is shown in Figure 6-22 and the other driving direction is shown in Figure 6-23. In Table 6-7 the results of the distribution for both drives are summarized. The distribution is zero mean with a good precision.

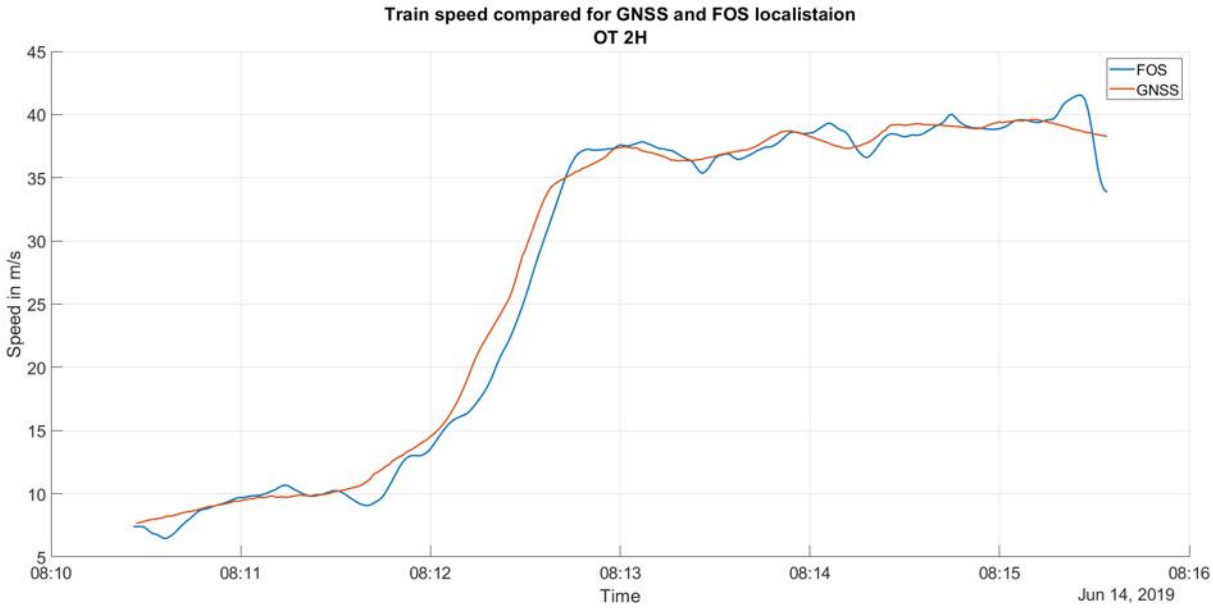

Figure 6-21 Train speed compared for GNSS / IMU and FOS for OT_2H drive with the measurement train.

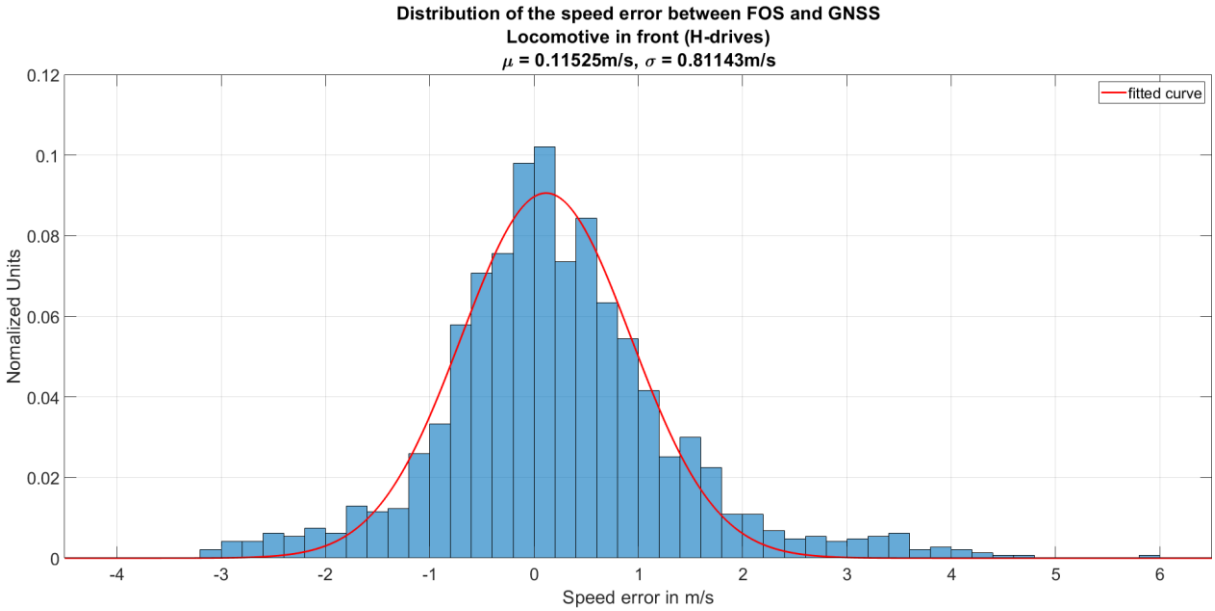

Figure 6-22 Distribution of the speed error between GNSS / IMU and FOS for the drives with the locomotive in the front position.

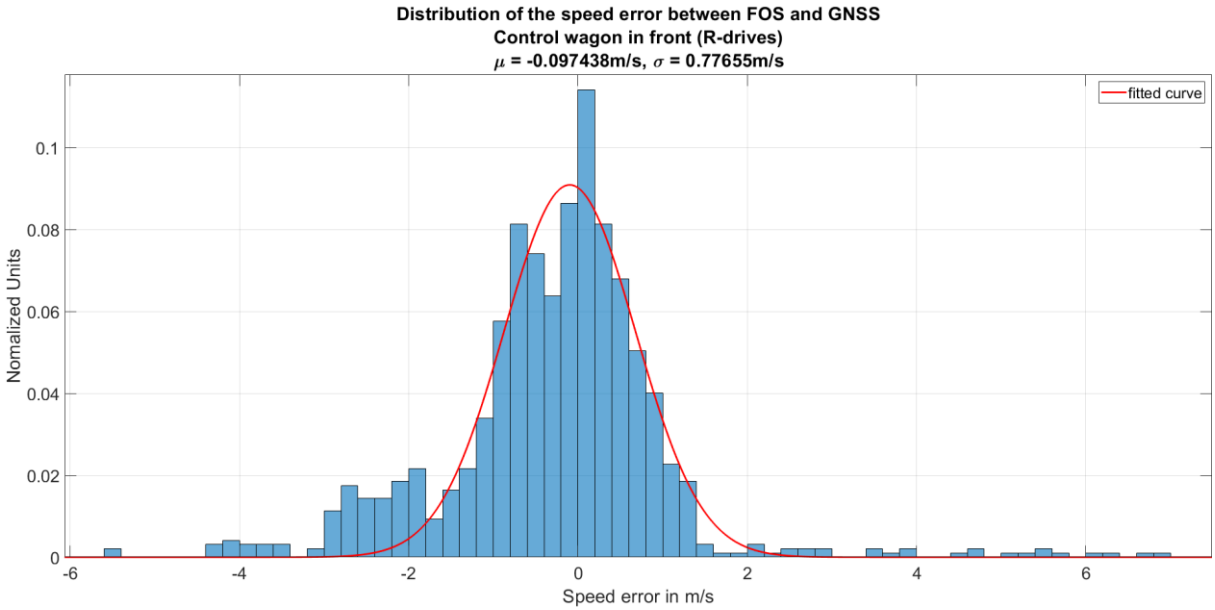

Figure 6-23 Distribution of the speed error between GNSS / IMU and FOS for the drives with the control wagon in the front position.

Table 6-7 Results of the gaussian fit of the distribution of the difference between the speed calculated with GNSS / IMU and the speed calculated with FOS. For both direction it is a nearly zero mean distribution. The precision is the width of the distribution.

Speed error	Mean error	Precision
OT_H	0.12m/s	0.81m/s
OT_R	-0.10m/s	0.78m/s

6.3.5 Comparison of Video, GNSS and FOS with Axle counters

Another ground truth was provided in the form of axle counter data for 5 axle counters. These axle counters' positions are accurately measured and also SIL 4 certified.

Unfortunately, the clock sources of the axle counters are not synchronized. To estimate the time delay, a data set that covered 1 hour was used which contained 17 trains. For each axle counter the delay was estimated separately because they did not have the same clock.

These time delays were used to evaluate the localisation error for the whole measurement data from 14 June 2019 (about 6:55 hours). Figure 6-24 shows the distribution of the linear distance (localisation error) between FOS and axle counter positions separately for each axle counter due to the different clocks. The corresponding summary of the results, listing the percentage of evaluation points for different error intervals, is shown in Table 6-8. The errors are significantly higher than in the GNSS / IMU comparison, indicating that these are not originating from the FOS measurements. Most likely, the larger errors are due to bad clock synchronization between FOS and the axle counters.

Figure 6-25 shows the time difference between axle counter clock (reference) and the clocks of the other sensors for each axle counter. GNSS, GNSS / IMU, Visual Odometry as well as FOS all show the

same pattern for the time difference so it is assumed that there were some issues with the axle counter time.

When looking at the section for axle counter ZP05211 in Figure 6-25 the time difference for both GNSS / IMU and FOS varies from drive OT_1R to OT_4R for about 2.5s. With a speed of 40m/s this results in a maximum error of 100m and therefore the results in Figure 6-24 are comprehensible. The distribution of axle counter ZP05111 is not shown in Figure 6-24 because his position was outside the good part of the mapping between channel and real world coordinates, which means that the hammering started at channel 96 and the position of ZP05111 is at channel 70 and here the real world coordinated were just extrapolated and so the results are not meaningful.

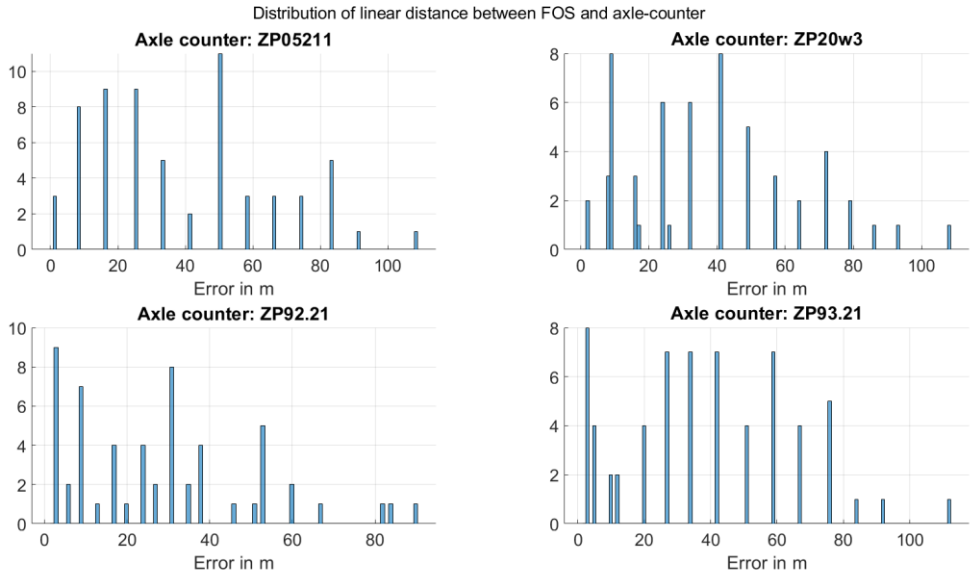

Figure 6-24 Distribution of the localisation error between FOS and axle counters.

Table 6-8 Percentage distribution of the localisation error when comparing axle counter and FOS. Table corresponding to Figure 6-24.

Localistion error	< 5m %	< 10m %	< 15m %	< 20m %	Min m	Max m
ZP05211	4.69	17.19	17.19	31.25	0.86	99.64
ZP20w3	3.51	22.81	22.81	29.83	1.56	108.04
ZP92.21	15.79	31.58	33.33	42.11	2.33	89.91
ZP93.21	12.5	21.88	25.00	31.25	2.38	111.43

Figure 6-25 shows the differences in the clock of the systems. The axle counter clock is taken as a reference. It looks like all the sensors have different clocks but the interesting thing is that GNSS, GNSS / IMU, Visual Odometry, and FOS all have a constant delay between each other for all drives. Only the distance to the axle counter clock varies. It seems like the clock changes with every drive, which means that the clock drifted with time.

Table 6-9: Time delay of the different sensor systems when passing the axle counter position. Reference time is the axle counter clock. Graphical representation of the table can be seen in Figure 6-25.

Sensor	Time (OT_1H)	Time (OT_2H)	Time (OT_3H)	Time (OT_4H)
Axle Counter MS ZP05111	08:53:55	10:10:26	11:26:07	13:38:15
Fiber Optic Sensing (FOS)	+5.9	+3.0	+5.3	+4.0
Visual Odometry (VO)	+4.8	+3.9	+4.0	+3.3
GNSS	+3.9	+2.7	+3.1	+2.1
GNSS + IMU	+3.1	+2	+2.1	+1.1
Axle Counter WCHZP20w3	08:55:33	10:13:23	11:28:25	13:40:58
Fiber Optic Sensing (FOS)	+4.9	+5.3	+3.8	+3.5
Visual Odometry (VO)	+3	+3.5	+2.4	+3.1
GNSS	+2.4	+2.4	+1.4	+1.4
GNSS + IMU	+1.6	+1.5	+0.5	+0.4
Axle Counter WCH ZP92.21	08:55:47	10:13:38	11:28:39	13:41:37
Fiber Optic Sensing (FOS)	+4.8	+4.8	+4.0	+2.8
Visual Odometry (VO)	+3.7	+3.5	+2.9	+3.6
GNSS	+2.5	+2.3	+1.6	+1.5
GNSS + IMU	+1.7	+1.4	+0.8	+0.4
Sensor	**Time (OT_1R)**	**Time (OT_2R)**	**Time (OT_3R)**	**Time (OT_4R)**
Axle Counter WCH ZP93.21	09:16:34	10:33:37	12:00:50	14:18:39
Fiber Optic Sensing (FOS)	+5.2	+4.2	+4.1	+2.9
Visual Odometry (VO)	Not available	Not available	Not available	Not available
GNSS	+2.6	+1.5	+1.4	+0.7
GNSS + IMU	+1.8	+0.7	+0.6	-0.2
Axle Counter MS ZP05211	09:18:20	10:36:21	12:03:28	14:21:41
Fiber Optic Sensing (FOS)	+5.7	+4.2	+4.1	+3.3
Visual Odometry (VO)	Not available	Not available	Not available	Not available
GNSS	+3.3	+2.4	+2.2	+0.8
GNSS + IMU	+2.5	+1.5	+1.2	-0.1

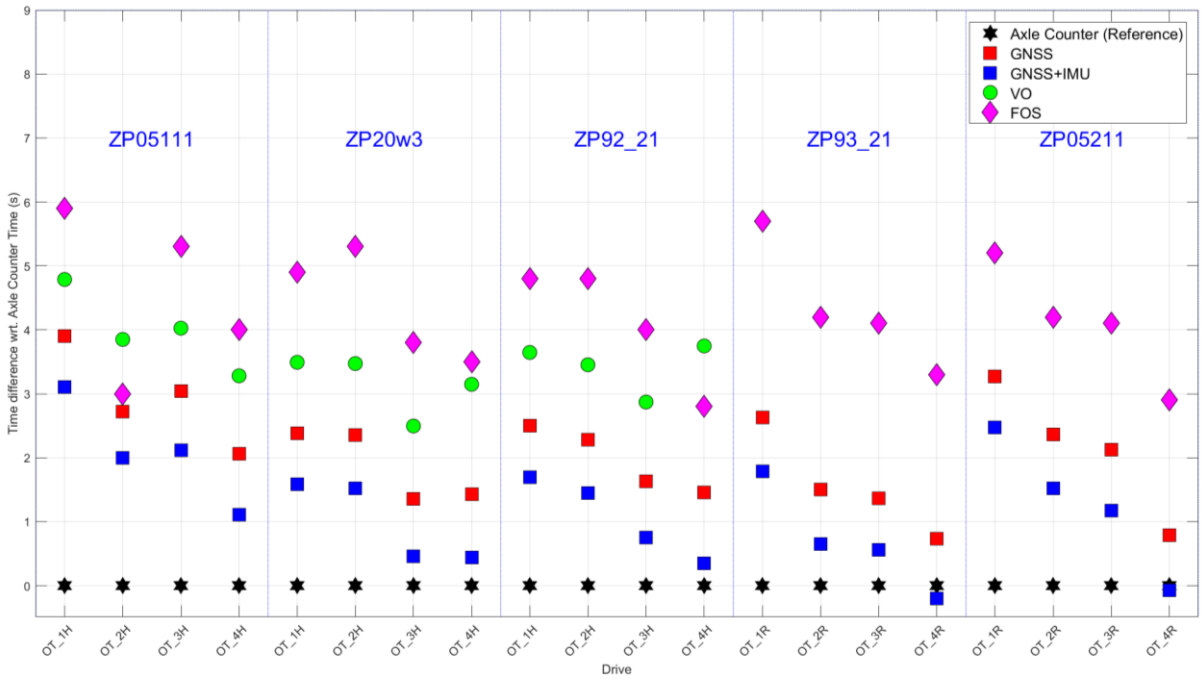

Figure 6-25 Differences in the clocks of the used systems. The axle counter clock is taken as a reference. The figure is divided into blocks which correspond to the axle counter written on the top of each block. On the x-axis the drives are plotted.

6.4 Conclusion

The comparison results of the different sensors/sensor systems show that the time synchronization (time stamps + latencies) is crucial for a multi sensor setup. The different sensor data has to be propagated to the same event time in order to be combined or compared (e.g. for voting the best sensor value). Techniques for the fusion of sensors with different sample properties are shown in [28].

6.4.1 Video

Due to their low cost and compact size, camera systems are in use in many applications across domains (automotive, robotics, …).

A simple system based on a mono camera with a 1280x1024 pixel resolution, operating at 60 Hz and pointing to the railway track, can collect data in real time and measure the train position with high accuracy.

Applying our highly innovative autocalibration procedure, which is based on the identification of the railroad track in the image, no further prerequisites are necessary and easy mounting of the camera system on the train's windscreen has been proved as well as a "plug-and-play" operation.

Visual Odometry

The relative motion of the train is measured by means of the so-called Visual Odometry. By comparing images between consecutive frames, the one-dimensional (1D) motion of the train can be determined.

The absolute distance travelled is calculated and compared to other sensor technologies like GNSS / IMU and GTG. The trueness is found to be less than 0.6%, depending on the measurement run analysed. The systematic error of the distance travelled, which is measured by Visual Odometry, is 0.8%.

Measuring by Visual Odometry the absolute distance travelled between consecutive couples of balises and comparing it to the distance stored in the database, which is considered as ground truth, the trueness is found to be less than 0.7%.

Based on the measurement runs, the train speed was calculated with Visual Odometry and compared to the combined measurement from GNSS and IMU, considered as ground truth. The trueness is found to be less than 0.7 km/h with a precision from 0.6 to 1.0 km/h, depending on the measurement run analysed.

The results are in accordance, within the given systematic uncertainty, between reference and measured data.

Furthermore, the 3D position of the train has been calculated by Visual Odometry, too. As expected, such a method can be very precise on a short scale, but it suffers of systematic uncertainties that accumulate over time causing a drift in the calculated position.

Video Localisation

Therefore, Visual Odometry is supported by Video Localisation in the calculation of the train absolute position. Indeed, the precision of the calculated train position is substantially improved by referring to infrastructure objects detected by the camera system, like point-frogs, that have a fixed and exactly known position. The gradually increasing drift is corrected every time the point-frogs are detected by the camera.

Thus, the 3D position of the train has been calculated and compared to GNSS / IMU measurements, considered as ground truth. The precision of the measurement with Video Localisation ranges from 21 to 26 cm, depending on the measurement run analysed.

In addition, or as an alternative to point-frogs, artificial landmarks like AprilTag can be used. All the AprilTags, located alongside the track, were successfully detected by a dedicated camera with a large focal length. According to the point-frogs, the fixed position of the AprilTags can be used to reset the drift of Visual Odometry.

Options and Restrictions of the current Approach

Currently, there are still some limitations of the camera systems in use, which are described in the following:

- Poor illumination in long tunnels: With the current camera system, it is not possible neither to identify the railway track nor to measure the train position in long tunnels with poor illumination. Extensions to the current system are planned in order to deal with poor illumination. The near-infrared illuminator can be replaced by one with higher power or it can be mounted outside, so that the light emitted will not be reflected by the windscreen.

- Weather conditions: Several additional measurements have been analysed to validate different use cases under different conditions that could limit the performance of the algorithms. The performance of the algorithms seems not to be affected by little snow on the railway track. The performance in challenging weather conditions like heavy rain or fog have not been tested yet, since they didn't show up in the measurement runs. New measurements shall be taken in order to evaluate the precision of the measured position in such weather conditions.

- Processing time of the algorithms: Images were stored in real time, while the calculation of the train position has been performed offline with real time capable algorithms. It shall be noted, that the scope of this analysis was to exploit the full potential of Visual Odometry in the measurement of the train position. A real time measurement of the train position is the final goal and can be reached in the next future by slightly tuning the parameters and the algorithms in use to fulfill the requirements of the train localisation.

In spite of the excellent results achieved in this PoC, the accuracy of the train position, measured by Visual Odometry, still has some room for improvement. The following list shows options based on the current approach:

- Fixed camera system: The accuracy of the train localisation can be simply improved by installing the camera system in a fixed position. A manual calibration with a calibration sheet (chessboard) would measure the camera initial pose as well as the absolute scale with higher precision than the automatic procedure based on the railway track identification.

- Larger focal length for the detection of railway objects in the infrastructure: The camera and the objectives were selected in such a way, so that one camera could detect AprilTags alongside the track and the other camera could track features in the surroundings as well as identify the railroad track. Nevertheless, the very good results of correcting the train position, calculated with Visual Odometry, with the fixed position of the detected point-frogs, suggests the use of already existing objects identified in railway infrastructure as global reference. Therefore, a camera with large focal length, combined with another camera with shorter focal length, both pointing to the railway track, is a very promising combination, since it would allow a good tracking of features in the surrounding as well as a detailed identification of objects within the railroad track.

- Camera: A camera with a higher resolution would increase the precision of the measurement of the train position. A camera able to collect color images could improve the identification of objects like balises. However, it shall be noted, that the number of pixels of the images affects the processing time.

- SLAM (Simultaneous Localisation And Mapping) for drift compensation: Currently, point-frogs are used to reset the accumulated drift. In addition, artificial landmarks like AprilTags are detected and can be used as well to reset the accumulated drift. The plan is to use a system that is able to map the position of objects like bridges, trees or buildings detected in previous runs. Then, the system shall also be able to re-identify those objects once they come into view after a train passes again, and use their position to compensate for the drift.

Extent of use in railways

The presented analysis reveals a huge potential for the Visual Odometry and Video Localisation as part of a future continuous, accurate and reliable train localisation.

The main advantage of the Visual Odometry is the high precision in the short range. As explained in the report, the systematic uncertainty can be lowered by using a fixed camera mounted on the windscreen and the precision of the absolute distance is determined by the pixel size, that are smaller than 1 cm. The combination of the travelled distance measured by Visual Odometry with the global position measured by GNSS, seems to be a very promising approach.

Regarding the determination of the 3D position of the train, the results presented show that a precision of about 20 cm can be reached by Video Localisation. The precision can be drastically improved by increasing the number of natural objects to compensate for the drift. By the introduction of SLAM algorithms, the number of natural objects along the paths can be detected and their position used for drift compensation. Thus, it could be used standalone for absolute localisation for dedicated use cases.

Furthermore, Video Localisation could be thought to generate Train Position Report (TPR) messages as an input for ETCS. Currently, the TPR is generated from the last passed balise (group) and the travelled distance from there. As shown in the report, the distance between balises can be measured accurately by Visual Odometry. This leads to another approach to introduce virtual balises. A virtual balise could be a point-frog or an AprilTag and it could replace the current balise for the generation of the TPR.

In addition to measuring the train position, images collected by a camera could give valuable information for other railway applications, and especially for infrastructure applications, e.g. for automatically detecting stopping plates alongside the track or for track monitoring and updating of the existing data bases by ongoing measurement of the track width, curvature and other parameters.

6.4.2 FOS

Fiber optic sensing offers great potential as a supporting technology not only for train localisation but also for train length and integrity, among others. It provides absolute positions, train speed and train length in real time, which makes it unique compared to other technologies. In addition to locating trains, this sensor can also be used to detect rockslides, animals or people on tracks. In this document only the train localisation was covered.

The analysis was architecturally divided into two parts. The first one is the intra channel analysis which is concerned with the processing of the signal coming from the interrogator unit (raw sampled vibration data) and the second is the inter channel analysis which receives the processed and thresholded data from the intra channel and models a moving train. We have used a simple thresholding model and only

one measure (ESF) for this report. A more complex model, with hysteresis thresholding coupled with both power and ESF for detecting silence and non-silence would produce better (less noisy) data for the inter channel analysis. The results, however, would still be dependent on the train speed and would produce results consistent with the data reported here. This doesn't seem to be a problem because it was modelled in the inter channel algorithms.

One usage for FOS would be as a fall-back option for train integrity determination. With our algorithms it was possible to recognize and track all trains in the given data compared to the train schedule we received. Also, two additional trains were recognized in the data from 14 June 2019.

In order to be able to make a statement about the train integrity, the length of all these trains along the journey was calculated and compared with the true length from the provided train schedule. The error between calculated length and real length lies with an interval of +-20m for about 87% of all the data points. There are some outliers where the calculated lengths are quite different from the ones given in the schedule. This should be further investigated and errors in the schedule should not be discarded.

The results regarding the use of FOS for train integrity determination are good but there is still a lot of things which can be improved to get even better results.

To evaluate FOS and our algorithms for train localisation the idea is to compare the results achieved by FOS with already certified sensors to prove that the new localisation system has the same performance regarding quality and safety. As was shown in section 6.3 there was a huge problem with the synchronisation of the clocks, especially for the evaluation with the axle counter. It was not possible to properly synchronise the clocks between FOS and the Axle counters. This presented a negative impact on the comparison and fixing this would definitely improve the results, i.e., the results reported are definitely worse due to the time discrepancy. In fact, the lack of a "master clock" for all sensors has proved to be a challenge in order to actually evaluate their relative accuracy and is a high priority for sensor fusion. Also, further measurement drives should take the various clock sources into consideration and target their synchronization. At the very minimum, each clock source should supply a confidence interval for its values in relation to a universal clock source.

However, comparisons with GNSS / IMU showed very promising results as it was possible to estimate the time difference of the clocks. The localisation error between GNSS / IMU and FOS can be interpreted as the estimation of the reference position of GNSS / IMU on the measurement train. For drives from Münsingen to Uttigen the locomotive was in front position and the estimated mean value for the reference position of GNSS / IMU was 36.57m with a precision of 7.73m. The real value of the position according to the train data is 34.18m. The difference to the estimated value is good when considering the 8m resolution of FOS.

For drives from Uttigen to Münsingen the control wagon was in front position and the reference position of GNSS / IMU is therefore 32.32m behind the front end. The estimated mean value was 27.95m with a precision of 7.13m. Again, the difference between real value and estimated value is within the resolution of FOS. The sum of both estimations is 64.52m and is an estimation of the train length, which is 66.5m.

Great results were achieved with FOS and the algorithms we implemented therefore when comparing to the point where we started. In the last month of the project great improvements were made in the used algorithms and models. But there is still a lot what can be done to get even better results.

There are some different models which can be tried for the tracking in the Inter Channel Analysis and also the signal filtering in the Intra Channel Analysis could be improved by analysing more data. We have a firm conviction that better results can be achieved especially in the area of train length determination. For most of the trains tracked, the length was measured with a high degree of accuracy over the entire measuring section. Some trains, however, presented a large discrepancy in relation to their lengths given in the schedule. These require additional investigation to find out the real cause of these discrepancies.

7 SBB innovation project - Optical Train Localisation

7.1 Introduction

Generally, computer vision algorithms can be grouped in two different categories, classical algorithms and machine learning based algorithms. The optical approach for exact train localisation presented in **Chapter 3** is based purely on classical computer vision algorithms, such as edge and line detection. The main reason is, that the process to reach a SIL4 certification, as required for train localisation, is currently not well established for machine learning based approaches. However, deep-learning algorithms, especially based on convolutional neural networks (CNNs), have led to a huge improvement in many computer vision areas, such as object detection and distance estimation. Further, machine learning based algorithms are already essential in the self-driving car industry and for selected cases, safety certification has already been approved.

In the following proof of concept (PoC), we investigate a deep learning based optical approach for exact train localisation. In the first iteration, we investigate the following use cases:

- Optical detection and recognition of tracks and selection of driven on track
- Optical detection of further objects of interest along the tracks
- Influence of lighting and weather conditions on the optical detection

Here we find, that the track selective lateral position of the train can be determined with a very high accuracy. For this we first detect all tracks in a frame a camera mounted at the front of the train. Note, that for most lighting and weather conditions the detection precision and recall are well above 90%. However, we note that during night or at low visibility, the detection precision drops below 70%. In general, the presented optical approach only works, if all adjacent tracks are visible in the front camera. We note, that the track detection can be impaired due to multiple reasons.

- Lighting and weather conditions (e.g. night or fog).
- Limited camera resolution or horizontal field of view (FoV). e.g. camera does not capture all adjacent tracks)
- Obstructed tracks. e.g. at train station entrances or for track covered by soundproof walls

Further, we find that optical detection of other object is also possible. However, sufficient training data is needed for a reliable detection.

In the second iteration, we investigate the following use cases:

- Integration of topology Database (DfA) with optical track selection to obtain a track specific localisation.
- Investigation of the robustness of the optical detection with respect to further lighting and weather conditions as well as for further routes.

Here we find that using a course GNSS signal it is possible to merge information from the DfA and the optical track selection to obtain a track specific train localisation. However, ‘matching’ is only possible if the optical detection of the tracks is complete, as in the event that optical detection and topology data do not match, the observation is discarded. Here, matching depends very much on the quality of the image acquisition (resolution, field of view, sensor noise, …). For this reason, the current algorithm only works on sections where all tracks are visible, and less than 5-6 tracks are present. Therefore, the developed algorithm currently only evaluates tracks outside station entrances and does not evaluate if track edges run over switches.

In the third iteration, we investigate the following use cases:

- Optical detection and recognition of mast boards to determine the longitudinal position of the train
- Optical detection and recognition of kilometer panels to determine the longitudinal position of the train
- Optical detection of switch state and expected driveway of the train

Here we find, that the alignment of the mast boards along the tracks leads to a poor recognition rate. We note, that this recognition rate can be improved by using a side-camera (45° or 90° relative to the direction of travel) together with image pre-processing algorithms. However, we observe a very good recognition rate for the kilometer panels (79.47 %). Further, we demonstrate that on a test route from Thun to Ostermundigen the optical kilometer panel detection, together with the optical track selection and the DfA integration, can be used without GNSS for full train localisation. Note, that the test route is approximately 20.4 km long and of the 204 kilometer boards 151 are fully captured in the camera images. On this test route we successfully detect 120 kilometer boards (79.47 %) and identify 91 track sections. This corresponds to a detected kilometer board every 170 m. However, around stations we observe longer non-recognized distances (up to 1.1 km) and consequently also shorter non-recognized distances for overland sections. Further, we present possible approaches to determine the longitudinal position more precisely using the kilometer panels. However, further data is needed to refine and evaluate these approaches.

For the optical detection of the switch state and expected driveway, we find that semantic segmentation can be used to select only the expected driveway, even when moving over switch sections. However, we note that the algorithm was only tested on a limited set of examples and has to be trained and tested using additional data.

In conclusion, we show that an optical train localisation at different lighting and weather conditions is possible, without the use of additional external infrastructure. Further, the optical approach demonstrated in **chapter 7**, does not exhibit scale drift and does not depend on an external signal. However, we note that the approach relies on the visibility of all adjacent tracks and kilometer panels. Here, the occlusion of tracks or masts as well as extreme lighting and weather conditions can lead to a failure of the described approach. This has to be especially considered for tracks around station entrances and at conditions with poor visibility, such as at night or in a tunnel. However, a different camera setup (night vision camera, wider FoV, different camera alignment, …) and additional infrastructure (e.g. additional tags at ambiguous sections) could quickly lead to a significant improvement. In general, more data is needed to refine and further evaluate the optical train localisation approach described in **chapter 7**.

The investigated PoC in **chapter** 7 is structured as follows. In **chapter 7.2** we describe the setup of the employed camera system. In **chapter 7.3** we present the 1st iteration of the developed optical localisation. In **Iteration 1,** we develop and evaluate a deep-learning based algorithm to detect train tracks and to select the used track in an image taken out of the front of the train (train driver perspective). Further, the performance of the track selection algorithm is tested for various environmental conditions, such as snow, fog and rain. Note that, the performance testing was also partially done in **Iteration 2** but is discussed here for consistency. In **chapter 7.4** we present and evaluate the 2nd iteration of the developed optical localisation. In **Iteration 2**, we use a course GNSS signal to estimate the longitudinal position of the train along the tracks. This positional estimate is used to extract the current track layout from the DfA. The track layout is then merged with the optical track selection algorithm to obtain a track precise lateral position of the train. In **chapter 7.5** we present and evaluate the 3rd iteration of the developed optical localisation. In **Iteration 3**, we replace the course GNSS signal, used in **Iteration 2** for the longitudinal position estimation, with an optical longitudinal localisation approach. Therefore, we develop a deep-learning based algorithm to detect and identify the km-sign posts along the tracks. This is done using a 2nd camera facing 45° degree relative to the direction of travel. The identified km-sign is then matched with the DfA to obtain the longitudinal position of the train. Note, that the km-sign posts appear

regularly and frequently along the tracks and that their position is already precisely documented in the DfA. Hence, the described optical localisation does not require additional external installations. Finally, we combine the lateral and longitudinal positioning approaches to obtain a stand-alone optical train localisation method. In **chapter 7.6** we discuss future building blocks and tasks required prior to the deployment of the presented optical localisation. **Chapter 7.7** summarizes the results and our conclusions.

7.2 Camera Setup

The developed optical localisation approach is based on dual camera setup mounted at the front of the train. The first camera, denoted as FRONT, points along the direction of travel and is used for the track selection. The second camera, denoted as TAG, points 45° degree relative to the direction of travel and is used for the km-sign detection and identification. The technical details of the used Speedgoat/M2C camera are described in **chapter 3.2.3** and the calibration parameters of the cameras are described in **chapter 3.2.5.** Note, that the approach also depends on access to an updated version of the DfA. However, the DfA, or the relevant section, can be stored locally in the OBU and doesn't require a permanent external connection.

7.3 Iteration 1

The goal of the first iteration of the presented Proof of Concept (PoC), which started in April 2018, was to answer the following fundamental questions.

- Can tracks be recognized and a track-selective position determined by capturing camera images and processing them by artificial intelligence or computer vision algorithms?
- Under what lighting and environmental conditions does the process work?
- Can qualitative and quantitative statements about the determination accuracy be made?
- What are the limits of the proposed approach?

In addition to the key questions about track selectivity, we also investigated the extent to which other road elements, that could possibly be used to determine the longitudinal train position, for example track-signals and balises, can be recognized by neural networks.

7.3.1 Track selective localisation

The basic idea of image-based track-selective localisation is to train a neural network to recognize all tracks in an image (see **Figure 7-1**). Next, the detected track layout is matched to the DfA to determine the track selective train position. Thus, in a first step we investigated whether it is possible to detect tracks with a high accuracy using an object detector based on the shape and extent of the tracks.

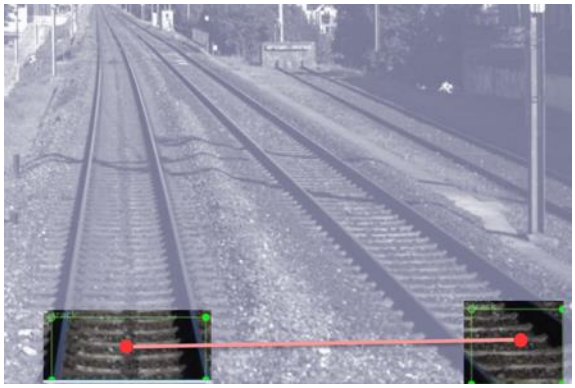

Figure 7-1. Example image of tracks detected using an object-based detector.

7.3.2 Image source

Currently there are already a number of different approaches in the literature to perform object detection in images. At the start of the PoC, the most widespread network architecture was the Region based Convolutional Neural Networks framework (R-CNN) as well as improvements derived from it, such as the Fast R-CNN. The R-CNN family of techniques primarily uses sub-regions in the image to localize the objects. This means that the network does not look at the entire image but only at the sections of the images which have a higher chance of containing an object.

In contrast, the YOLO framework (You Only Look Once), deals with object detection in a different way. The YOLO network takes the entire image in a single instance and directly predicts the bounding box coordinates and class probabilities for these boxes. Further, the image is scaled three times to improve the detection of small objects. This approach also explains the high number of network layers (see **Figure** 7-2) in the YOLO network architecture. The biggest advantage of using YOLO is its detection speed – it is incredibly fast compared to other CNNs and can process 45 frames per second on standard hardware. Additionally, it is one of the best algorithms for object detection and has shown a performance comparable to the R-CNN algorithms (see **Figure 7-3**).

For these reasons we chose the YOLO framework for the initial Iteration of the presented PoC.

However, note that further network architectures with similar or slightly better precision and with accelerated inferencing have also been developed and are continuously being developed. If the PoC is continued, we advise to reinvestigate the choice of the object-based track detector.

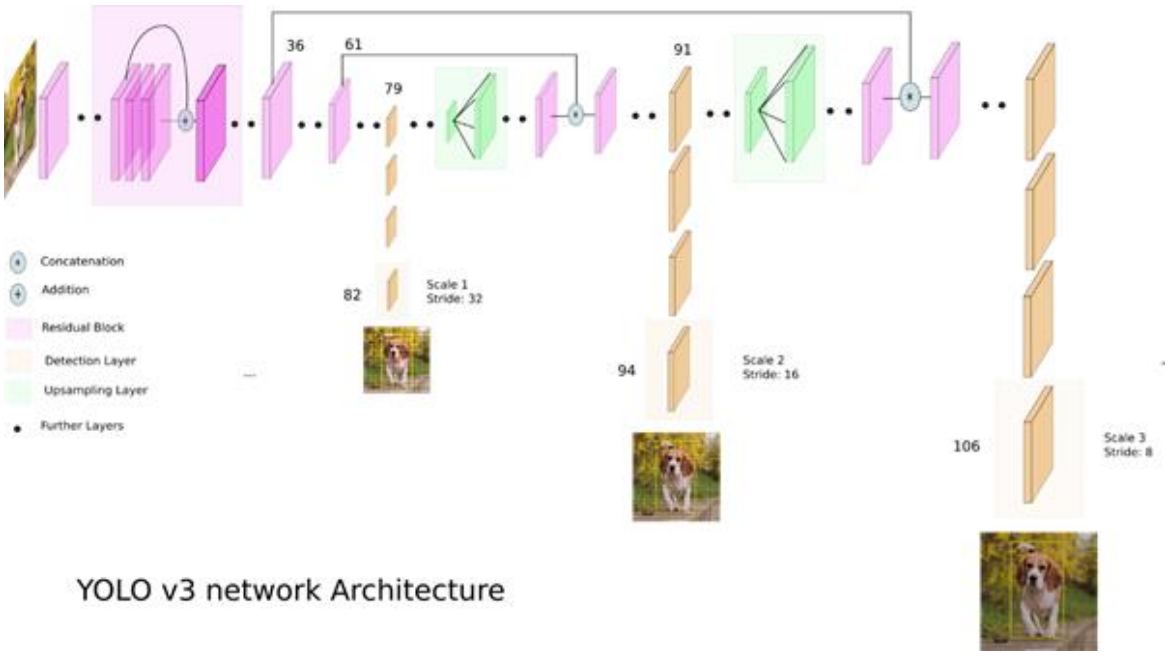

Figure 7-2**.** Schematic representation of the YOLO framework architecture.

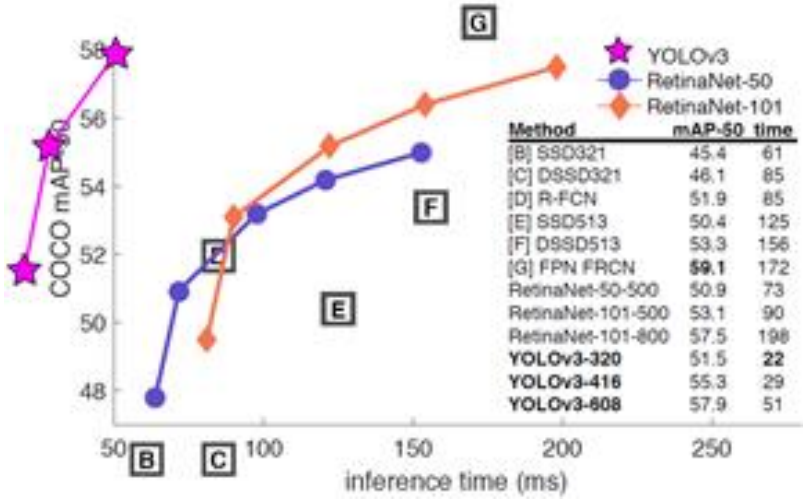

Figure 7-3. Mean average precision (mAP-50, higher is better) shown against the inference time (ms) for different CNN architectures (YOLO framework and different R-CNN frameworks). The mAP-50 is calculated on the COCO object detection dataset from Microsoft.

7.3.3 Image pre-processing

No explicit steps for image pre-processing were required for the training. We deliberately used the image material in different native qualities and resolutions.

Additionally, YOLO has several internal image augmentation algorithms that can be randomly applied to the images during training. This includes ‘angle’ for the rotation of the image, ‘saturation’ for changing the color saturation, ‘exposure’ for adjusting the brightness and ‘hue’ for changing the hue.

Due to extensive augmentation, the data set for training can be massively expanded and usually leads to a better generalizable more accurate trained neural network.

7.3.4 Labeling of tracks, signals and other objects

Prior to training and evaluation, all used image data has to be correctly labeled in order to generate the ground truth, against which the accuracy of the network prediction is measured. This means, that in every image the position and the class of all the objects, which should be detected, have to recorded.

For this purpose, we have labeled training and evaluation data extracted from the DFZ at different lighting and environmental conditions. For the actual labeling, we used the Python based open source tool ‘labelImg’. However, we adapted and expanded this to support and accelerate the labeling (see **Figure 7-4**).

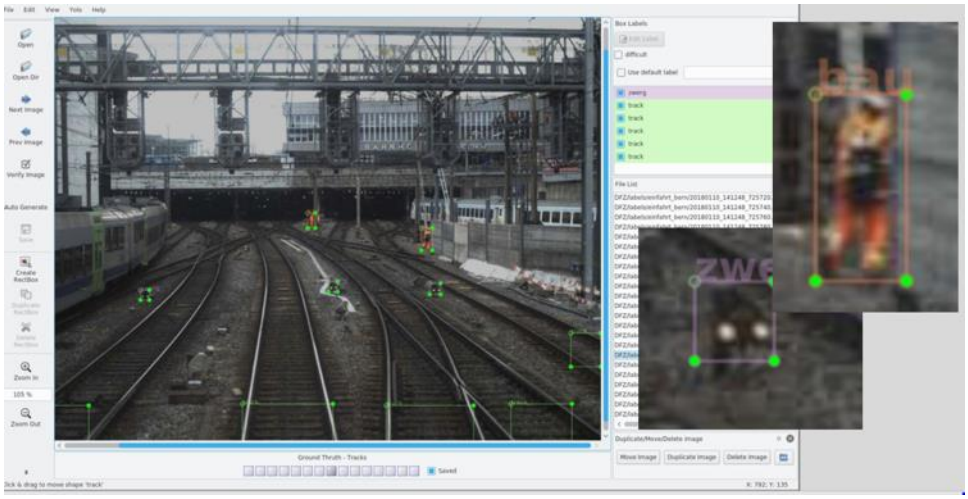

Figure 7-4. Adapted interface of the Python based open source tool 'labelImg'. The left image shows an example image extracted from the DFZ. The two images on the right show zoomed in sections of the main image, showing a detected person and "Zwergsignal", in order to draw the object bounding boxes more precise.

Ideally, the YOLO framework should be trained with at least 2000 labels per object class. In view of the high effort involved in labeling and the high recognition rate achieved, we have temporarily used a lower number of labels for some of the object classes.

In total, we labeled training and evaluation data from 2520 images with 6 classes, resulting in 7917 labels in total and with a strong focus on the class 'track' (see **Table 7-1**).

Table 7-1. Object classes and number of labels per class in the generated training and evaluation data.

Class	# Labels
Balise	646
Person	157
Km	394
Signal	740
Track	5487
Zwerg	493
Total	7917

7.3.5 Training

The generated dataset was split as follows; 80% of the labeled data was used for training and 20% for validation. Training was carried out on a Nvidia DGX-1 system, which is equipped with 8 GPU's. However, only 4 GPU's were used for the training, as this is a limitation of the network implementation used with YOLO. Note, that there are pre-trained YOLO networks available based on the COCO dataset from Microsoft. However, we did not use transfer learning during training and instead trained the network from scratch.

The training took place in two steps. In the first step, 1000 batches are carried out on a single GPU for initialization. Only in a second step, the network is then trained on several (up to 4) GPU's. During training, we observed GPU loads of up to 300 watts per GPU. By using 4 GPUs instead of one, the training time was reduced from 5 days to 1 day.

The development of the values for true positive (TP), false positive (FP) and false negative (FN) detections during training turned out as expected (see **Figure 7-5**).

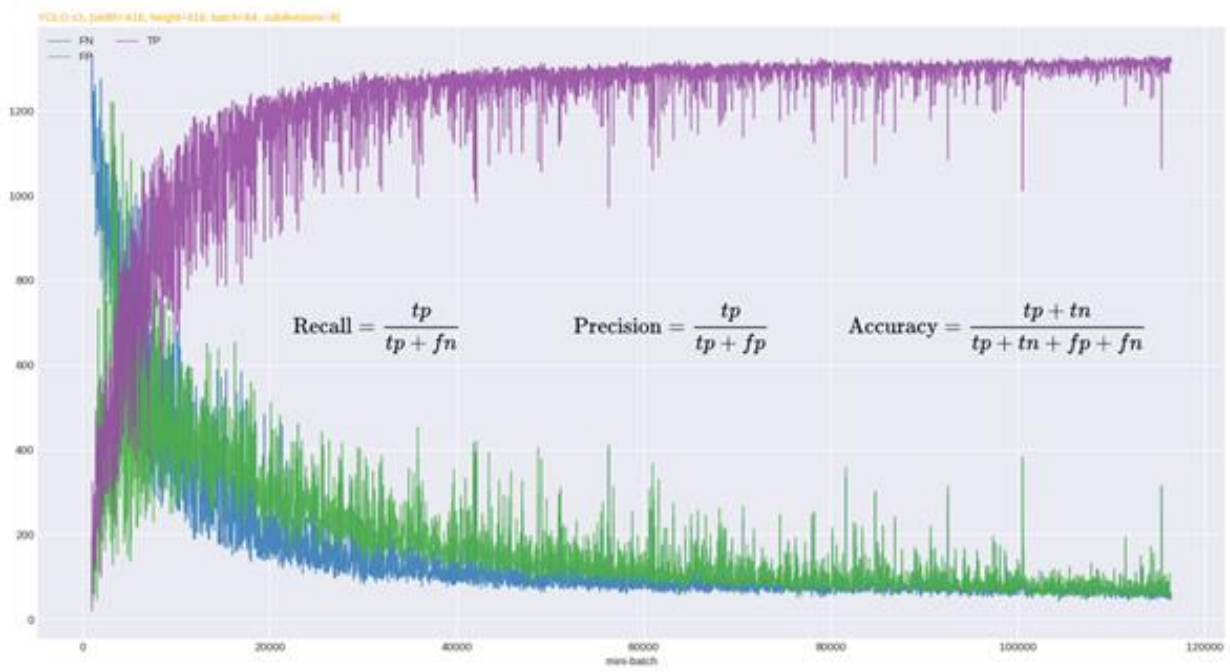

Figure 7-5. Development of the values for TP (purple), FP (green) and FN (blue) detections during training. As expected, the TP detections increase, and the FP and FN detection decrease towards a limit during the training process.

7.3.6 Validation

In order to validate the trained network, we used the 20% validation images together with the labeled “Ground Truth” for the tracks. For these images we defined and marked which tracks are visually recognizable (see **Figure 7-6**). For each image in the validation set the detected tracks were then compared with the marked tracks of the “Ground Truth” to calculate the precision and accuracy. Images were only counted as true positive (TP), if all tracks in the image were fully recognized.

Figure 7-6. Example of validated image containing detected tracks **(top)** and “Ground Truth” labels **(bottom)** for visible tracks. The example shows a false positive (FP) detection (left box), where a side-rail is erroneously detected as track.

Table 7-2 shows the results grouped according to the different investigated lighting and environmental conditions. For the performance of the track detection algorithm we calculate the number of TP (correct detection) the number of FP (wrong detection), the number of FN (object not detected), the accuracy ($\frac{TP}{TP+FP+FN}$, the fraction of correctly detected objects), the precision ($\frac{TP}{TP+FP}$, the ratio of correctly detected objects over all detected objects) and the recall ($\frac{TP}{TP+FN}$, the fraction of correctly detected objects over all objects that should have been detected). Generally, a high recall means that most of the objects are detected, whereas a high precision means that most objects are detected correctly. Further, we also show the F1 score, which is the harmonic mean of the precision and the recall.

In general, for all lightings and environmental conditions, except during night, we achieve a precision and a recall above 90%. However, note that for some of the conditions the precision is higher than the recall. This means, that the algorithm sometimes misses the tracks but if a track is visible, then the algorithm can detect it with a high accuracy. Further, the somewhat poorer performance at night, recognizable by the higher number of false positive detections, is due to, among other things, the strong color noise of the camera. Better results can certainly be achieved here by using more sensitive cameras or using infrared (IR) cameras, possibly together with IR headlights.

In summary, it is particularly interesting here, that in addition to the optimal conditions during the day and with sunshine, the tracks can also be recognized at night and in fog or even while covered with snow (see **Figure 7-7** and **Table 7-2**).

Table 7-2. Evaluation results of the track detection algorithm for different lighting and environmental conditions. The rows “two-lane track” and “chiasso” are representative for track detection during good weather conditions.

Dataset	TP	FP	FN	Accuracy	Precision	Recall	F1
tunnel	714	9	59	0.98	0.98	0.92	0.95
two-lane track	320	1	35	0.98	0.99	0.90	0.94
fog	5108	88	49	0.99	0.98	0.99	0.98
snow	2661	44	226	0.98	0.98	0.92	0.95
night	5313	2300	278	0.95	0.69	0.95	0.80
dusk	3869	29	243	0.99	0.99	0.94	0.96
chaisso	794	12	60	0.98	0.98	0.92	0.95

Figure 7-7. Examples of track detection at different lightings and environmental conditions (from top left to bottom right: snow, cloudy, tunnel, dusk, fog and night).

7.3.7 Development of GUI for demonstration

For the first iteration we additionally developed a simple graphical user interface (GUI) to demonstrate the results to the stake holders. Since we expect high performance requirements and, on the other hand, wanted to keep the technical effort within limits, we did not develop the GUI with web technologies but instead based on Python and Qt. Qt is a graphic framework that is written in C++ and delivers an outstanding performance via a 'signal' based event system. There exists also a stable and tested Qt language binding for Python. Nevertheless, the effort for developing a GUI is considerable. Especially when high demands are placed on the performance.

With the future development of the presented PoC in mind, the goal is, that the GUI should be able to handle two video channels with 60 frames per second each. In the first iteration, we were able to achieve a processing speed of approximately 35 FPS for one video channel. This is mainly due to the large number of layers in the YOLO v3 network. Note, that further optimisations and adjustments are necessary here and will be discussed in the following iterations.

7.4 Iteration 2

7.4.1 DfA Topology database

In the first iteration of the presented PoC, we showed that it is possible to use optical methods to recognize tracks, even under difficult light and weather conditions. However, recognizing the tracks and the siding alone is not enough to determine a position. Thus, in the continuation of the PoC, in the second iteration, we investigate to what extent the optically obtained information can be used for a position determination. The first step is to clearly determine the specific track, i.e. the track on which the train is traveling.

The key to this is the derivation of the data from the SBB topology database “DfA, database of fixed systems (Datenbank der festen Anlagen)”.

- The DfA is based on the national coordinate system LV95 with the axes E-East and N-North with seven-digit coordinates (Bern = 2,600,000 / 1,200,000).
- It consists of a track and route network
- The track network is a directed graph whose nodes are called "*Turnout points*” (Weichenpunkt) and the edges are called the "*Track lines*” (Gleisstrang). A switch point always belongs to a switch and a track is always connected to a turnout point.
- The length development in meters is defined as a metric on each track. It has the value 0 at the start point and “Track length” at the end point.
- The length development is calculated and mapped as “*Track points*” at fixed intervals (10m) and stored in the DfA database.
- All objects and derived points are georeferenced.

Access to the DfA, which takes place via standard SQL, is very complex in terms of queries, since in addition to the existing system, configured and dismantled systems are also saved. For this reason, we initially loaded, preprocessed and stored the data required for our purposes in a separate data format. This makes it possible to completely load the topology relevant for the PoC into the memory. Due to this approach, the access latency to objects is considerably shorter (see **Table 7-3**).

Table 7-3. Used DfA objects with corresponding load time in seconds.

Object	Items	Load Time s.	Description
tile2Object	117291	0.52	Maps objects to tile
Gleisstrang_gleispunkt	70084	0.19	Maps Gleispunkte for a given gleisstrang
strecke	3013	0.01	Operation lines
betriebspunkt	3330	0.01	Operation points
weiche	28069	0.23	Switchers
weichenpunkt	77414	0.28	Switch points
gleisstrang	74925	0.24	Track lines
gleispunkt	1103114	14.25	Track points
streckenpunkt	464126	3.98	Operation point
mast	152748	3.43	References to the poles

In order to localize objects in the DfA, a geo-position is necessary. Currently (in iteration 2) This is not available for purely optical processes. We therefore use the geo positions of the DFZ test drives that are available as meta-data of the images in this phase of development. From these geo positions, we can derive an approximate position of the train. For this purpose, we convert the geo position to a tile position. The conversion is based on the algorithms of the “OpenStreetMap” open source project [https://wiki.openstreetmap.org/wiki/Slippy_map_tilenames#Python]. This also enables us to use the OpenStreetMap tiles for map visualization.

To map the objects, the geocoordinates of the DfA objects "*Track-lines"* and "*Track points*" are connected in a first step and the tile coordinates are determined (see **Table 7-4**).

Table 7-4. Extracted and calculated DfA objects for a determined coordinate tile.

Object	Attributes	Calculated
Gleisstrang 'Track line'	**id** id_weichenpunkt_beginn id_weichenpunkt_ende id_gleisstrangart gleisstrang_bezeichnung laenge ...	
Gleispunkt 'Track point'	Id **id_gleisstrang** id_strecke y, x, z stationierung ...	All track points for each track string in a determined coordinate tile are calculated.

7.4.2 Topology matching process

The process to compare and match the optical observations with the DfA topology data takes place in 3 steps:

A.

1. An image of the front camera is analyzed by the YOLO network. Detected tracks are determined by a bounding box with their position and size.
2. The horizontal arrangement of the tracks is determined using the x and y coordinates of the bounding boxes.
3. Since the camera has a fixed attachment point, the track on which the train is running is therefore determined constantly. Further, the positions of the bounding boxes on the x-axis are used to determine whether a track is to the left or right of the current lane.
4. By evaluating all track detections, a "track trace pattern" can now be created. In the example shown in **Figure 7-8,** the extracted "trac trace pattern" is given as '111'.

B.

1. The tile coordinates are determined on the basis of a course GNSS signal of the train.
2. The connections from track point to track point of the tracks that are in the tile coordinate are calculated and drawn on the map (green lines **Figure 7-8 B**).
3. The direction of travel of the train is determined on the basis of previous positions.
4. A straight line is now placed perpendicular to the direction of travel (red line **Figure 7-8 B**).
5. The intersections of the straight lines between the track points and the straight line perpendicular to the direction of travel are determined.
6. The track trace pattern of the intersection points, in the example '111', which result from the DfA topology is determined.

C.

1. A matcher algorithm now receives the track trace patterns determined by steps A. and B. and determines a possible match.
2. If the patterns match, the position of the tracks on the left and / or right is determined based on the direction of travel. Otherwise, no position is determined for the current frame.
3. It is determined on which track, e.g. Left / Right / Middle etc., the train is running.

4. With the track determined, the specific track section can now be determined in the topology. In the example «154512 - FMUE 16.1 - GTS 21.1»

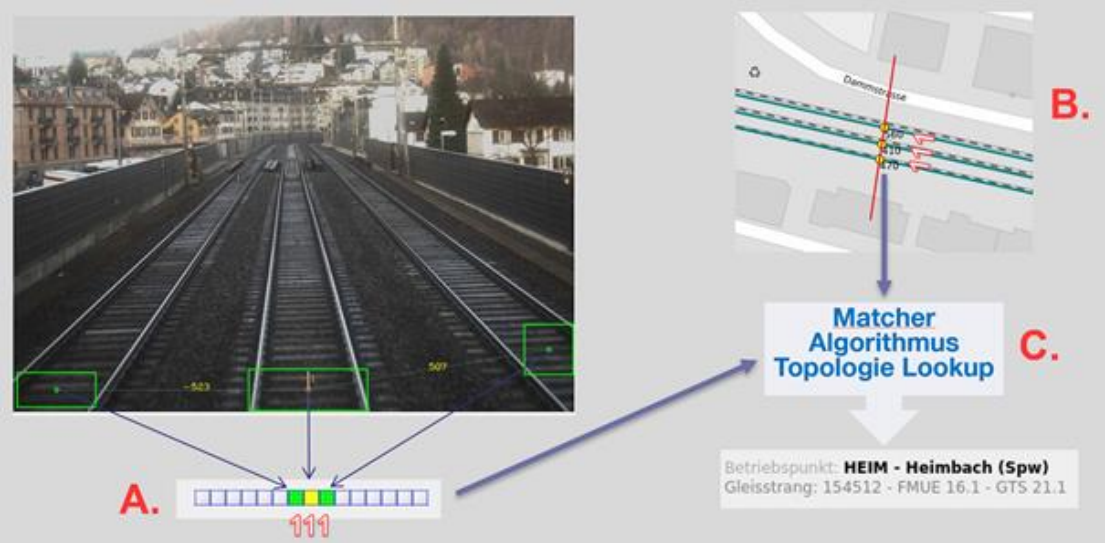

Figure 7-8. Example for topology matching process to determine the track selective train position. **(A)** From the image of the front camera the visible tracks are detected (green boxes) using the YOLO network and the track trace pattern (111) is determined. **(B)** The connection between the track points within the determined coordinate tile are calculated and drawn on the map (green lines). Then, using a line perpendicular to the direction of travel (red line), the DfA track trace pattern (111) is determined. **(C)** Finally, the matcher algorithm compares the two tracks trace patterns and determines the track selective train position.

Note that, ‘matching’ is only possible if the optical detection of the tracks is complete. This depends very much on the quality of the image acquisition. The resolution, horizontal opening angle of the optics, sensor noise etc. have a strong influence on the complete track detection. In the event that optical detection and topology data do not match, the observation is discarded. This usually occurs on multi-lane lines (> 5-6 tracks), tracks covered by soundproof walls or in train station entrances. Currently, the developed algorithm only evaluates tracks outside station entrances and does not evaluate if track edges run over switches.

We also note that, the internal data structures used in the presented PoC map the topology via python dictionaries and lists for efficient access. The geometric calculations for determining the intersection point are carried out via Qt and are implemented with high performance in C++. Code profiling shows that the majority of the time is used for object detection by YOLO. Thus, the matching does not have a measurable influence on the achievable frame rate.

7.5 Iteration 3

In the two previous iterations we described an approach to optically assign the lateral train position to a specific track. However, the approach still depends on GNSS to determine the longitudinal position, i.e. to extract the correct geo-position tile from the DfA. In the third iteration, we investigate the possibility to detect and recognize the kilometer and mast panels that are attached along the tracks, in order to determine the corresponding geo-position tile in the DfA. This would make the optical approach fully independent of a GNSS signal.

The kilometer panels are usually attached to the catenary masts, but can also be placed on the side of the floor or on walls etc. The panels attached to the masts represent a very precisely measured reference point, since the bases of the masts often serve as a reference for measurements during construction work. In the DfA topology database, practically all masts have georeferenced information. They are usually spaced at a distance of approximately 50 m and kilometer panels are attached to every 2nd mast. Thus, we have a fixed reference point every 100 m.

AprilTags are another alternative for recognizing a mast with a “QR code”. AprilTags can be read using classic computer vision methods and are thus recognized very robustly. A fundamental disadvantage is, that the tags would have to be attached to the entire route network, whereas the kilometer panels are already installed. The combination of both methods, kilometer panels on the tracks and AprilTags in train station entrances, possibly also for the track selective localisation, may be a useful addition.

7.5.1 Detection and recognition of kilometer panels

We re-trained the YOLO network used for the track detection (see **Chapter 7.2**) to additionally detect the kilometer panels from the front camera image. After the kilometer panels are detected, the recognized area is cropped from the unscaled image and passed on to another network for optical character recognition (OCR) analysis.

In our first experiments, we evaluated pre-trained networks, which were trained with data from the “Google Street View House Numbers SVHN” dataset. However, there was an insufficient recognition rate for digits on the kilometer and mast boards. House numbers are apparently too different. Therefore, we created our own training data set from the DFZ images and trained them with a 2nd YOLO network. **Table 7-5** denotes the different training classes and the number of instances in the DFZ training data set.

Table 7-5. Classes used for OCR analysis and the digits of instances in the DFZ training data set.

Class	# Labels
0	401
1	424
2	437
3	333
4	362
5	318
6	355
7	356
8	410
9	545

7.5.2 Training data

Extensive images were created as part of several test drives in June 2019 on the route from Thun to Ostermundingen. The image data required for our purposes were captured by a front camera and a side camera set up at a 45° angle. The images were captured in grayscale with 10-bit depth and 60 FPS. For training and inferencing, we have converted the image depth from 10 to 8 bits.

7.5.3 Km Boards

In order to recognize the full kilometer panel based on the individual digits, we developed an algorithm that brings the recognized digits of a board into a semantic context. This is done based on the bounding box positions and areas of the detected digits. Here, the kilometer panels have a fixed format. The kilometer is in the first line. In the second line are the hectometers, followed by the meters in a smaller font (see **Figure 7-9**).

Figure 7-9. (left) Image from a side camera pointing 45° relative to the direction of travel. The image shows a geo-referenced mast with a detected kilometer panel (red box) and a detected mast board (purple box). **(right)** Detected kilometer panel used for OCR analysis. The yellow boxes show the detected digits.

To evaluate the detection accuracy, we created a tool that can generate the “Ground Truth” for the kilometer panels, so that recognition rate per kilometer panel can be measured (see **chapter 7.5.8**).

Note, that there are various possibilities to make the algorithm more robust. For example, the ratio between the area of the whole kilometer panel and the area occupied by the detected digit can be used to determine whether all digits have been detected. If this is not the case, the choice of the YOLO architecture has to be re-investigated, as small digits might not be optimally recognized. Additionally, the use of classic computer vision algorithms is also conceivable. Further, the detection could be made more robust by combining several subsequent detections and by shifting the image section in the x and y direction to match the corresponding digits.

7.5.4 Catenary mast boards

In addition to the detection of kilometer panels, we also investigated if mast boards can be recognized (see **Figure 7-9**). As a rule, these are attached to the masts lengthways, or laterally in newer sections of the route. During the test drive in June 2019, additional pictures were taken with a camera oriented perpendicular to the direction of travel.

During the analysis, we were able to gather the following insights:

- The narrow surface of the mast sometimes results in extreme image contrasts during the image capturing, so that the image can only be made visible by applying extensive image corrections, e.g. Gamma correction, brightness, masked histogram and more (see **Figure 7-10**).

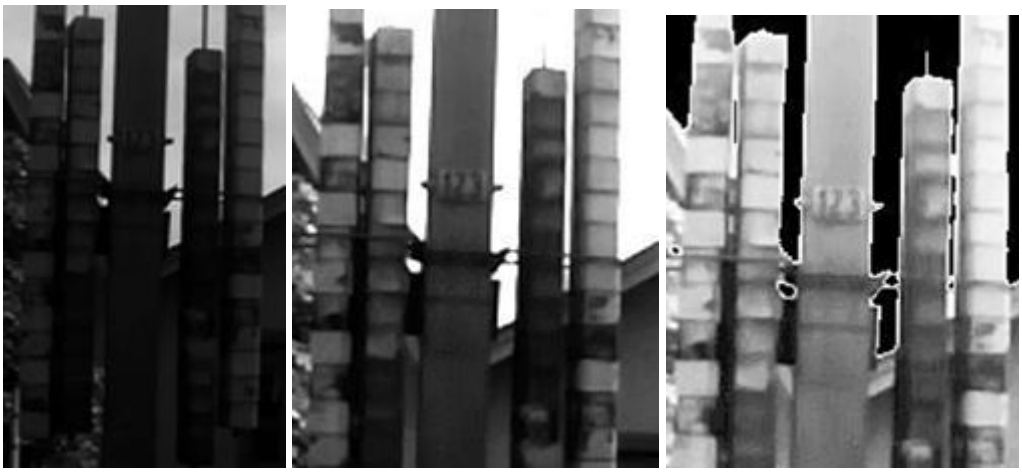

Figure 7-10. Example image from mast with extreme image contrast **(left)**. Effect of gamma correction on example image **(middle).** Effect of brightness and masked histograms on image gamma corrected image**(right)**.

- There is a possible strong motion blur depending on the speed. This is particularly pronounced in the case of boards that were recorded from a shorter distance. We tried to reduce the motion blur with “Wiener Deconvolution”, which was partly possible. However, in order for this algorithm to deliver optimal results, information about viewing angle and speed is required. This can only be achieved by great effort with more sensor data like IMU and odometry.

Figure 7-11. Example image of mast board with motion blurr **(left)** and after the application of a Wiener Deconvolution filter **(right).**

- With the side camera aligned at 45°, the mast panels can be read much better , although there is still the problem of the motion blur at higher speeds.

7.5.5 DfA Topology database

In addition to the objects «Track line» and «Track point» used in iteration 1, the DfA also contains objects that enable the a «Route» concept. There is also data for the assignment of masts, which are georeferenced, along the route.

- The route network is a directed graph. The nodes are called "Betriebspunkt Kilometrierung - operating point mileage", the edges are called "Strecke - route".
- Track routes that are on the same ballast bed are combined into one route.
- Route points are generated as 10 meter points, analogous to the track points on the track axis, and saved in the DfA.
- Each route has a start and end kilometer.

7.5.6 Position determination

In order to be able to assign kilometer information to the masts of a route, we have created a graph in which the nodes correspond to the kilometers and the edges to the routes. If a board is now visually recognized, the nodes of the kilometers and the edges with the routes are read in the graph. If there is only one edge, then the route has been found and the mast from the DfA has been read over the kilometers.

The position of the train can then be determined via the geo-reference of the mast and the lateral track position. This is possible because distance information from the middle of the track to the outside of the track bed is recorded for each track. Since the sizes of the plates and the number height as well as the parameters of the camera used are known, it is possible to improve the distance measurement even further with classic computer vision methods.

If there are several edges, i.e. several routes, we try, if a position has already been determined, to exclude the lines that are at a distance >1000 meters from the last position. If there is only one route left, the route has been found.

If there are no previous position determinations, we have to wait for the next Km table and make a new determination. As a rule, the mileage of the routes after a certain number of positions gives a clear profile. In the event that this is not possible, i.e. the routes do not differ at all, this uniqueness can be restored by attaching an additional kilometer board to one of the next masts.

When comparing the mileage on the board and the information stored in the Dfa, it is noticeable that there are some boards where the values do not match. We solved this problem with a mapping table, in which the kilometer information can be corrected manually.

With the procedure described here it is possible to get a position even without GNNS.

7.5.7 GUI

In order to be able to visualize the results of the methods described above, we have expanded the GUI from the first iteration. The video channels of the front and side cameras can now be displayed synchronized. The optical detection of the tracks and boards, the determination of the mileage of the boards and the algorithm for route detection run in real time (see **Figure 7-12**).

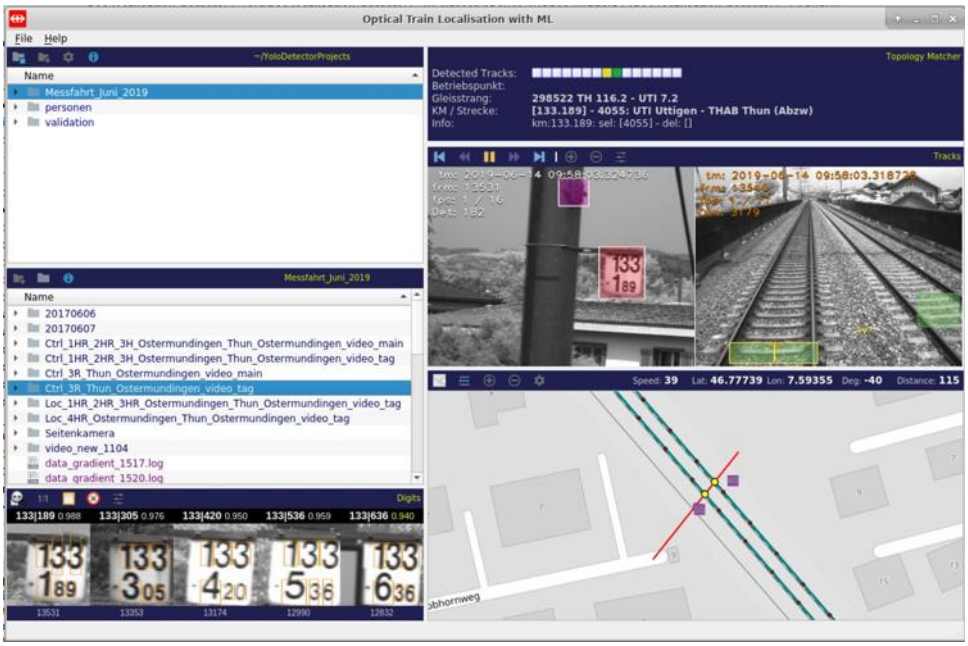

Figure 7-12. GUI of combined optical localisation approach up to the third iteration. The top right image shows a frame of the front camera with the detected tracks (green and yellow boxes). Above, the extracted track trace pattern is shown. The middle image shows the corresponding frame of the side camera (45°) of the mast and the detected kilometer panel (red box) and mast board (purple box). The detected kilometer panels are then used for OCR analysis (bottom left image). The yellow boxes show the detected digits. Finally, the extracted optical information is matched with the DfA data-base (bottom right) to extract the current position (top panel).

7.5.8 Analysis of the test data

As part of various test drives, images were recorded with the front, side and rear cameras in June 2019. For our analysis we used pictures of the trip “Ctrl_3R_Thun_Ostermundingen_video”. In **Figure 7-13** the map on the left shows the recognized kilometer boards along the route, the one on the right shows the track sections on which the train was traveling. Note, that at beginning of the trip a number of KM boards are needed first, until an initial position can be determined. Also note, that only the images, without any GNSS, were used to determine the train position.

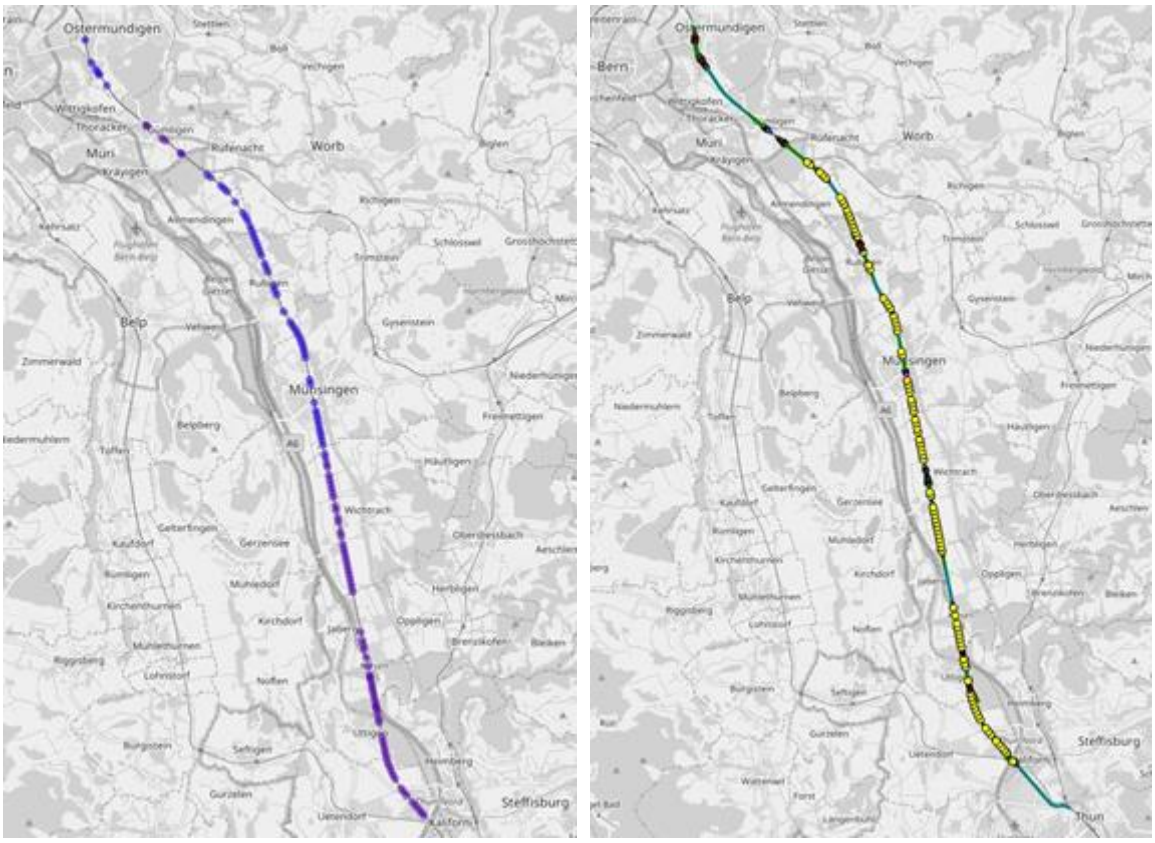

Figure 7-13. Analysis of recognized kilometer boards. **(left)** Recognized kilometer-boards along the route. **(right)** Traveled track sections extracted from the DfA (ground truth).

Table 7-6 shows the kilometer boards and track sections that have been recognized along the entire route.

Table 7-6. Recognized kilometer boards (blue) and track sections (black) along the route “Ctrl_3R_Thun_Ostermundingen_video” for the different route sections (purple).

1	UTI	Uttigen	-	THAB	Thun	Abzw.	(4055)		
133.961									
133.845									
133.741									
133.636									
133.536	TH	116.2	-	UTI	7.2	(298522)			
133.420	TH	116.2	-	UTI	7.2				
133.305	TH	116.2	-	UTI	7.2				
133.189	TH	116.2	-	UTI	7.2				
133.073	TH	116.2	-	UTI	7.2				
132.957	TH	116.2	-	UTI	7.2				
132.841	TH	116.2	-	UTI	7.2				
132.725	TH	116.2	-	UTI	7.2				
132.435	TH	116.2	-	UTI	7.2				
132.319	TH	116.2	-	UTI	7.2				
132.203	TH	116.2	-	UTI	7.2				
132.087	TH	116.2	-	UTI	7.2				
131.971	TH	116.2	-	UTI	7.2				
131.856	TH	116.2	-	UTI	7.2				
131.744									
131.632									
131.518									
131.402									
131.286									

2	KI	Kiesen	-	UTI	Uttigen	(332)			
130.894	UTI	2.1	-	7.1	(269603)				
130.800									
130.658									
130.542									
130.484									
130.428									
130.314	UTI	2.2	-	WCH	19.2	(269602)			
130.198	UTI	2.2	-	WCH	19.2				
130.082	UTI	2.2	-	WCH	19.2				
129.976	UTI	2.2	-	WCH	19.2				
129.870	UTI	2.2	-	WCH	19.2				
129.754	UTI	2.2	-	WCH	19.2				
129.700	UTI	2.2	-	WCH	19.2				
129.640									
129.520									
129.410	UTI	2.2	-	WCH	19.2				
129.290	UTI	2.2	-	WCH	19.2				
129.180	UTI	2.2	-	WCH	19.2				
128.950	UTI	2.2	-	WCH	19.2				
128.830	UTI	2.2	-	WCH	19.2				
128.720	UTI	2.2	-	WCH	19.2				
128.600	UTI	2.2	-	WCH	19.2				
128.500	UTI 2.2 - WCH 19.2								

3	WCH	Wichtrach	-	KI	Kiese	(331)			
127.900	UTI	2.2	-	WCH	19.2	(269602)			
127.669	UTI	2.2	-	WCH	19.2				
127.553	UTI	2.2	-	WCH	19.2				
127.437	UTI	2.2	-	WCH	19.2				
127.321	UTI	2.2	-	WCH	19.2				
127.209	UTI	2.2	-	WCH	19.2				
127.093	UTI	2.2	-	WCH	19.2				
126.977	UTI	2.2	-	WCH	19.2				
126.865	UTI	2.2	-	WCH	19.2				
126.749	UTI	2.2	-	WCH	19.2				
126.633	UTI	2.2	-	WCH	19.2				
126.517	UTI	2.2	-	WCH	19.2				
126.401	UTI	2.2	-	WCH	19.2				
126.291	UTI	2.2	-	WCH	19.2				
126.175									
126.059									
125.954	WCH	9.2	-	19.1	(290861)				
125.856	WCH	9.2	-	19.1					

4	MS	Munsingen	-	WCH	Wichtrach	(330)			
125.578									
125.472									
125.370									
125.270									
125.154									
124.922	MS	26.2	-	WCH	2.2	(221678)			
124.806	MS	26.2	-	WCH	2.2				
124.690	MS	26.2	-	WCH	2.2				
124.574	MS	26.2	-	WCH	2.2				
124.458	MS	26.2	-	WCH	2.2				
124.342	MS	26.2	-	WCH	2.2				
124.226	MS	26.2	-	WCH	2.2				
124.127	MS	26.2	-	WCH	2.2				
124.025	MS	26.2	-	WCH	2.2				
123.877	MS	26.2	-	WCH	2.2				
123.771	MS	26.2	-	WCH	2.2				
123.539	MS	26.2	-	WCH	2.2				
123.423	MS	26.2	-	WCH	2.2				
123.308	MS	26.2	-	WCH	2.2				
123.203	MS	26.2	-	WCH	2.2				
123.103	MS	26.2	-	WCH	2.2				
122.987	MS	26.2	-	WCH	2.2				
122.871	MS	26.2	-	WCH	2.2				
122.755	MS	26.2	-	WCH	2.2				
122.639	MS	26.2	-	WCH	2.2				
122.416									

5	RUB	Rubigen	-	MS	Munsingen	(329)			
121.925	MS	21.1	-	7003.2	(271448)				
121.808	MS	4.1	-	7003.1	(271447)				
121.693	MS	4.1	-	7003.1					
121.461									
121.345									
121.229									
121.113	MS	2.2	-	RUB	11.2	(250548)			
121.004	MS	2.2	-	RUB	11.2				
120.893	MS	2.2	-	RUB	11.2				
120.781	MS	2.2	-	RUB	11.2				
120.669	MS	2.2	-	RUB	11.2				
120.557	MS	2.2	-	RUB	11.2				
120.445	MS	2.2	-	RUB	11.2				
120.331	MS	2.2	-	RUB	11.2				
120.215	MS	2.2	-	RUB	11.2				
119.880	MS	2.2	-	RUB	11.2				
119.763	MS	2.2	-	RUB	11.2				
119.650	MS	2.2	-	RUB	11.2				
119.417	RUB	4.1	-	11.1	(128548)				
119.301	RUB	4.1	-	11.1					
119.186	RUB	4.1	-	11.1					

6	GUES	Gumligen Sud	Abzw.	-	RUB	Rubige	(2375)		
118.782	RUB	4.1	-	11.1	(128548)				
118.665									
118.550									
118.434									
118.318									
118.202									
118.086	GUE	51.2	-	RUB	2.2	(137514)			
117.970	GUE	51.2	-	RUB	2.2				
117.850	GUE	51.2	-	RUB	2.2				
117.740	GUE	51.2	-	RUB	2.2				
117.620	GUE	51.2	-	RUB	2.2				
117.510	GUE	51.2	-	RUB	2.2				
117.340	GUE	51.2	-	RUB	2.2				
117.230	GUE	51.2	-	RUB	2.2				
117.110	GUE	51.2	-	RUB	2.2				
116.770	GUE	51.2	-	RUB	2.2				
116.650	GUE	51.2	-	RUB	2.2				
116.540	GUE	51.2	-	RUB	2.2				
116.430	GUE	51.2	-	RUB	2.2				
116.310	GUE	51.2	-	RUB	2.2				
116.200	GUE	51.2	-	RUB	2.2				
115.857	GUE	51.2	-	RUB	2.2				
115.741	GUE	51.2	-	RUB	2.2				
115.222	GUE	51.2	-	RUB	2.2				
115.012	GUE	51.2	-	RUB	2.2				
114.898	GUE	51.2	-	RUB	2.2				
114.844									
114.619									
114.512									
114.343									

	114.240 113.744 113.704 113.620 112.246 GUE 5.1 - OST 56.2 (125491) 112.017 GUE 5.1 - OST 56.2 111.905 GUE 5.1 - OST 56.2 111.850 111.742 111.529 110.914

The test route from Thun (kilometer 131.961) to Ostermundigen (kilometer 111.529) is 20.43 kilometers long and contains 187 visually observable kilometer panels, which corresponds to one panel every 109 m. Note, that the route formally should contain 204 kilometer panels. However, 17 panels are not visible from the current camera perspective. Of the 187 visually observable kilometer panels, 36 are not fully in the camera FoV and are thus not detectable. However, this can easily be address with a different camera setup. For the remaining analysis, we thus use the 151 fully detectable kilometer panels as benchmark (100 %). On the full test route, 81 track sections and 136 kilometer panels (90.06 %) were detected. For the whole test route, we observe a maximum distance of around 1.1 km, where no kilometer board is successfully detected. However, we note that this is occurs mostly around stations and for overland sections, we observe maximum non-recognized distances of around 200 - 300 m. Of the not-detected kilometer boards, 9 (5.96 %) were wrongly detected and 4 (2.65 %) were not readable due to motion blur. Further reasons for non-detected kilometer boards and track sections during the journey are:

- Kilometer boards not completely visible (36 of 187 visually observable panels = 19.25 %) or not detectable due to motion blur (4 of 151 visually detectable panels = 2.65 %).
- The side camera was mounted in a horizontal orientation during the test drive. Since the KM boards are not always fastened at the same height, not all boards were completely recognizable during the test drive. This could be improved by vertically aligning the camera and expanding the field of view accordingly.
- Some kilometer boards were not correctly recognized by the neural network (9 of 151 visually detectable panels = 5.96 %). For good results, approximately 2000 labels should be recorded per class (0..9), ie approximately 20000 labels. Our training data set only includes approximately 4000 labels. The results can certainly be improved here by further labeling.
- In DfA Topology, a distinction is made between the track section of a mainline track and the track section of a switch. At the moment we are only evaluating tracks from mainline tracks. Switches could be processed in a further developed version of the PoC.
- Several kilometer boards are required before a first track can be determined. If we also use GNSS, this would be ideally possible with the first kilometer board.
- Tracks are not visually recognizable because they are too far on the outside of the picture. Tracks to the left or right of the lane are covered by walls, noise barriers, platforms, earth walls, etc.

7.5.9 Performance

With the following optimisations it would be possible to improve the inference performance:

- Reduction of the neural network from RGB to grayscale, i.e. from 3 to 1 channel
- Use of the Tiny-YOLO network architecture. In contrast to YOLO v3, this network has considerably fewer layers.
- Multi-threading in the Qt application for image processing tasks
- Multi-threading when accessing the YOLO networks

On the development system (Nvidia DGX-1), a performance of >400 FPS for two video channels could be achieved when using only one graphics processing unit (GPU). The load on the GPU is approximately 85%. It should therefore be possible to run the software on a lower performance system, e.g. Nvidia AGX-Xavier to operate in real time. There are also further conceivable optimisations, such as the use of TensorRT.

7.5.10 More precise lateral localisation

In the current approach the lateral position is determined through the detection of the kilometer panels. However, detection of the panels only gives the lateral position as “in front of the mast”. Here, we propose to use optical methods to determine the distance between the detected panel and the camera. Together with the track selective information the distance can then be used to accurately determine the longitudinal position of the train.

The exact distance between the train and the kilometer panel can be detected using different approaches.

- Stereo-vision distance estimation using the front and the side (45°) cameras. Note, that this approach requires the panel to be detected simultaneously in both images and the cameras have to be precisely calibrated and rectified towards each other.
- Distance estimation based on the size of the kilometer panel (see **Figure 7-14 top**). The kilometer panels have an exactly defined shape and size. Thus, the distance between the camera and the detected panel can be estimated based on the size of the panel in the captured image. Here, the size of the panel in the image can be measured using conventional computer vision approaches. Note, that this approach requires only one camera. However, the relative alignment between the camera and the ground plane has to be known.
- Mono-camera distance estimation based on deep-learning (see **Figure 7-14 bottom**). Currently there exists various network architectures, which can estimate pixel-wise depth in single camera images. However, the accuracy of these deep-learning approaches critically depends on the similarity of the training images and the prediction image. This means, that the networks have to be trained using the same camera setup and similar environments as for the final prediction. Here, two general types of approaches exist.
    - Supervised approach: The networks are trained using “Ground Truth” distance data extracted from Light Detection and Ranging (LiDAR) sensors.
    - Unsupervised approach: The networks are trained using only a stereo-camera setup or using sequential images. Note, that for the sequential images, the relative scale of the distance estimation is not given and has to be determined.

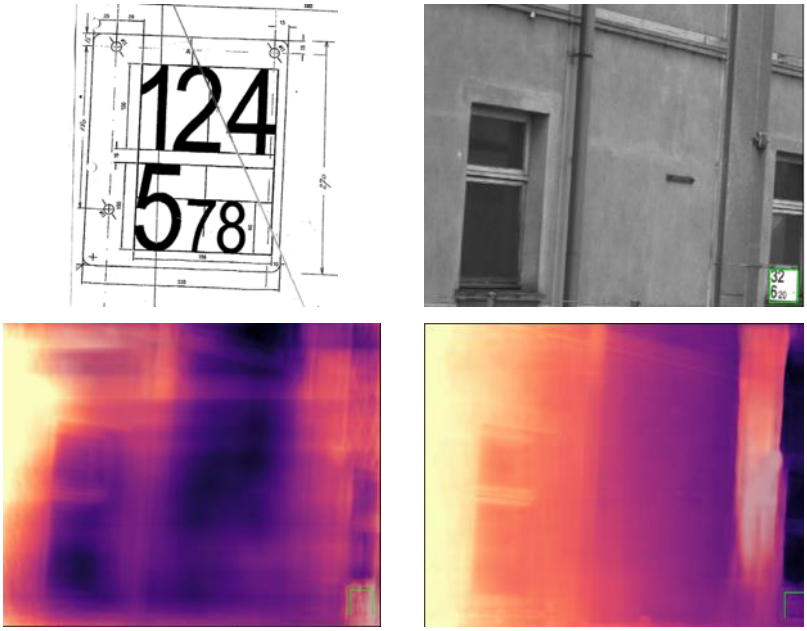

Figure 7-14. (top left) Shape and size regulation of kilometer panels. **(top right)** Image from side camera (45°) showing a mast and a detected kilometer panel (green box). Using the size of the kilometer panel, the distance is estimated as 24.5 meter. **(bottom left)** Pixel wise depth estimation using a pre-trained unsupervised network (Monodepth2 [*Godard, C., Aodha, O.M., Firman, M., Brostow, G., arXiv:1806.01260*]) for monocular depth estimation. Note, that the pixel wise depth is not well estimated and the distance to the panel is calculated as only 5.5 meter. **(bottom left)** Pixel wise depth estimation using an unsupervised network (Monodepth2) finetuned on DFZ data. Note, that the pixel wise depth is much better estimated than for the pre-trained model and the distance is calculated as 16.0 meter.

Note, that further experiments and parametrization is required to evaluate the accuracy and precision of all the possible approaches. This is discussed in detail in **chapter 7.6.1.**

7.5.11 Route prediction and improved track selection

In the current PoC framework the tracks are selected using object detection (see **Chapter 7.3**). In this chapter, we investigate the use of semantic segmentation to improve track selection. Additionally, semantic segmentation could possibly be used to not only detect the tracks but to also determine the switch position and to predict the future route. In contrast to object detection, semantic segmentation does not detect objects in the image, but instead determines the class of every individual pixel within the image. In the autonomous driving industry this technique is used to improve object detection, to determine the drivable area and for lane detection.

Here, we investigate the use of semantic segmentation for two tasks. First, we use semantic segmentation to detect all tracks in a given image. Second, we investigate the harder task of segmenting only the current drivable track. This step also involves detecting the correct pathway at visible switches.

Datasets. For the semantic segmentation of all visible tracks we used the RailSem19 dataset [*O. Zendel, M. Murschitz, M.Zeilinger, D. Steininger, S. Abbasi, C. Beleznai: **RailSem19:** A Dataset for Semantic Rail Scene Understanding. CVPR Workshops 2019: 32-40*]. This dataset contains 8’500 images (1920 x 1080 px) from front perspective of the train from all over the world. Additionally, the dataset already contains ground truth labels for the rails (*rail-raised*) and the track bed (*rail-track)*. Note, that we expect the publication of the RailSem20 dataset, which should contain a more diverse set of track scenes.

For the pathway prediction, we adapted 273 images of the RailSem19 dataset to only segment the current drivable track. Additionally, we used 191 images (640 x 360 px) from the SBB (Passenger Traffic) internal education videos for train drivers, to generate the SBBSem19 dataset. For this dataset we selected images containing interesting switch and track layouts. We then used the track selection model trained on RailSem19 to segment all visible tracks in the SBBSem19 dataset. Finally, we manually deleted all non-drivable tracks from the segmentation ground truth.

Network architecture. For both semantic segmentation tasks we use a CNN based on the DeepLab framework, which is currently the state-of-the-art for semantic segmentation. We also investigate the use of the Harmonic DenseNet architecture, which promises an accelerated inference. However, we found that the DeepLab framework achieves a better performance at around the same inference time as the DenseNet architecture.

Training. The training of the semantic segmentation network is done as follows:

1. Pre-train network using the Cityscape dataset (25’000 annotated images, including segmentation mask of 30 classes, of urban street scenes). For this step, we downloaded an already pre-trained network directly from the publishers.
2. Transfer learning using RailSem19 dataset to detect all visible tracks (only use *rail-raised* and *rail-track* segmentation classes).
3. Transfer learning using adapted RailSem19 and SBBSem19 datasets to detect only the current drivable track.

Note, that the second step can also be omitted for the route prediction. However, including the second step leads to a better performance. **Table 7-7** shows the hyper-parameters used for all of the transfer learning steps.

Table 7-7. Hyper-parameters used for the transfer learning steps.

Image size	Batch size	Learning rate	Momentum	Iterations
640 x 360 px	88	7E-3	0.9	100’000

Results. In general, a good performance is achieved for both segmentation tasks. For the route prediction task, we find, that a better performance is achieved if a single combined label instead of two separate labels for *rail-raised* and *rail-track* is used. **Figure 7-15** shows the route prediction results for an exemple set of images. Note, that the route prediction works at different lighting conditions and for different switch and track layouts. However, we note that the evaluation is not extensive and has to be done in a more qualitative manner for a wider range of lighting and environmental conditions as well as for more diverse switch and track layouts. For example, the current dataset contains only a very limited number of cross-switches.

Additionally, the current approach does not contain any short-term memory. Thus, if the train is located exactly above a switch, the correct route is ambiguous and cannot be predicted.

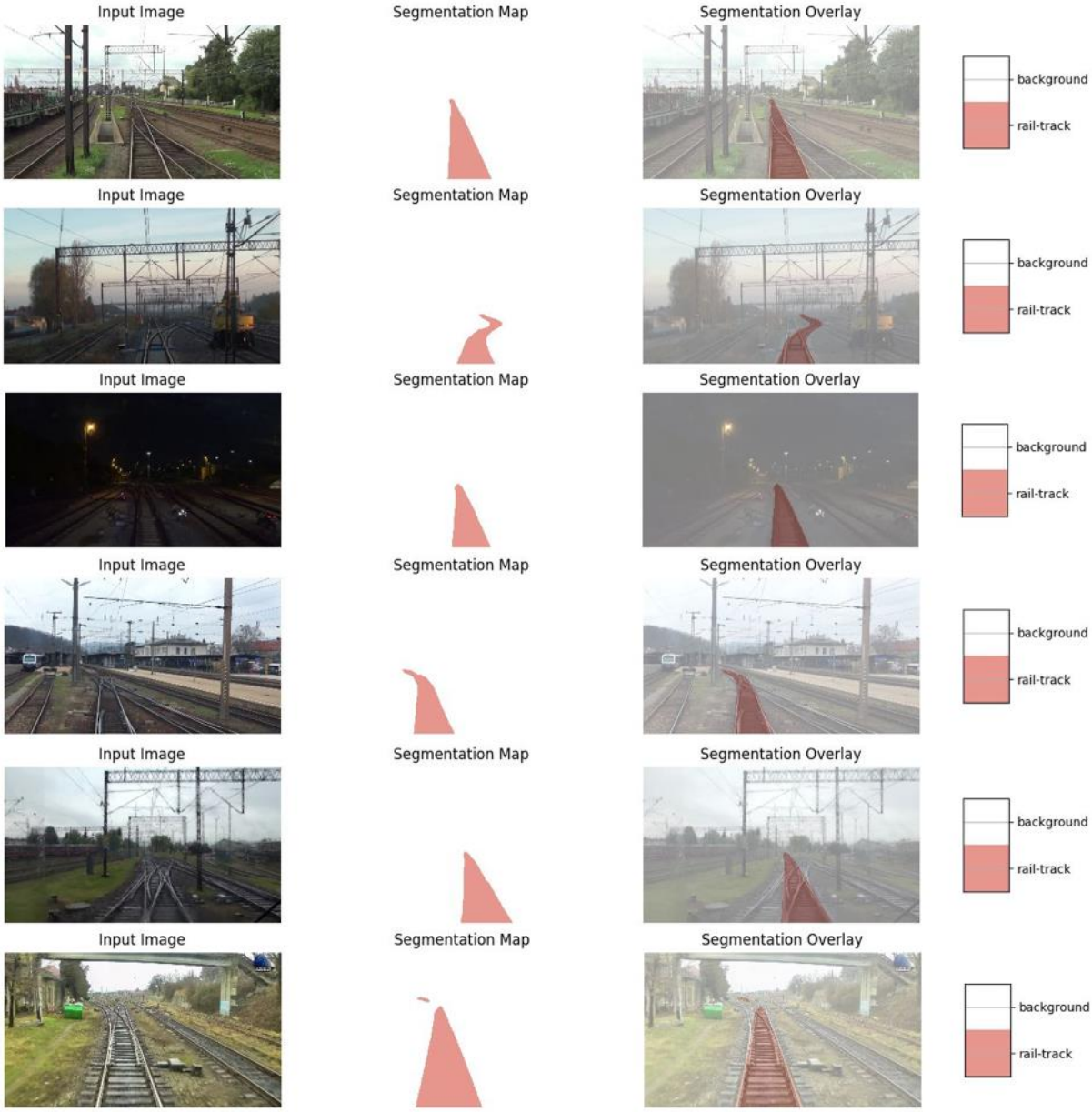

Figure 7-15. Example images showing the semantic segmentation of the predicted route.

7.6 Next Steps

The next building blocks required, prior to the deployment of the presented optical localisation are roughly sorted into four tasks. Whereas their priority is given according to the order below.

- Further development and refinement of the optical localisation method
- Data processing, standardization and publication (SBB intern)
- Evaluation and direct comparison of optical localisation with the other investigated localisation approaches (optical flow, GNSS, IMU, odometry, FOS)
- Sensor fusion (optical localisation, optical flow, GNSS, IMU, odometry, FOS)

7.6.1 Further development and refinement

Moving forward there are three aspects of the optical localisation, which have to be improved or developed further.

- Night vision and restricted visibility
- Route prediction and improved track selection
- More precise lateral localisation

Night vision and restricted visibility. Optical localisation is well suited to act as a complementary method to other localisation approaches, such as GNSS or IMU, as it is not limited by the same disadvantages (required external signal or scale drift). However, optical localisation has its own limitations, which have to be addressed prior to deployment. The main limitation is its failure due to restricted visibility conditions, such as fog, blinding light or at night. In order to overcome these limitations, we propose to investigate the use of short-wave infra-red (SWIR) cameras. Currently SWIR cameras are mainly used for military applications due to their ability to produce high quality images even at restricted visibility conditions, such as blinding light, fog and at night. For commercial applications the use of SWIR has been mainly limited by its high cost and the highly inconvenient SWIR camera setup. However, recently commercial SWIR cameras based on CMOS chips, such as the Raven SWIR camera from TriEye (https://trieye.tech/), have become available at a greatly reduced price range. Further, SWIR cameras can be combined with active illumination in the infra-red spectra, without affecting the train driver or other involved people.

As a next building block, we propose to investigate, if such SWIR cameras can be used in the presented optical localisation approach.

Route prediction and improved track selection. In the current approach all the tracks at the bottom of the image are detected using deep-learning based object detection. The used track is then selected trough knowledge of the relative camera position on the train. This approach has a high success rate for up to four roughly parallel tracks. However, there are many sections in the swiss railway network, which have a more complex track layout. This is mainly the case in the vicinity of railway stations. In such sections there are often more than four tracks simultaneously in the camera field of view (FOV) and the tracks are not parallel but contain many crossings and intersections.

Here, we propose two possible solutions. First, more training data of such track layouts could help to boost the detection performance. Second, the detection of the switch states could aid the lateral localisation. The idea is, that starting on a known track, e.g. coming from a simple track layout, knowledge of the switch states can help to track the exact lateral position of the train. Further, detection of the switch state would enable optical route prediction. This could also be beneficial for other applications.

Here, we suggest investigating two different approaches to detect the switch state. Note, that there might also be other approaches available in the literature. The first approach consists of using deep-

learning based object detection to directly determine the different switches and their states. The second approach is based on the lane-detection approaches used in the autonomous car industry. Initially, semantic segmentation is used to detect the selected track throughout the whole image. Next, the switch state is determined based on the position of the detected switch and the geometric layout of the segmented track.

Within our data set, we are already able to reliably detect switches, without their state, and to successfully segment the selected track through the whole image (see **chapter 7.5**). However, both approaches do not yet generalize well to other switch and track layouts. Thus, we need to acquire more training data of different switches to boost the performance of the algorithms.

More precise lateral localisation. In the current approach the lateral position is determined through the detection of the km-signs. However, detection of the signs only gives the lateral position as “in the vicinity of the sign”. Here, we propose to use optical methods to determine the distance between the detected km-sign and the camera. Together with the lateral train position the distance can then be used to accurately determine the longitudinal position of the train.

In **chapter 7.5** we present different optical approaches to measure the distance based on our camera setup. However, all of the presented approaches require additional data. The geometric based approaches require the known coordinate transformation between the front and the side camera. Further, they also require the known coordinate transformation between one of the cameras and the ground-plane. This can either be achieved using an initial calibration together with an IMU or using ground-plane estimation based on the observed tracks (see **chapter 3.3**). Additionally, all approaches require ground-truth distances from LiDAR, either for training (deep-learning based approaches) or for evaluation (all approaches)

Further, we propose to extend the lateral localisation approach to other objects, such as switches, which are well referenced in the DfA.

7.6.2 Data processing, standardization and publication (SBB intern)

Currently all the generated raw and labeled data (detected tracks and other objects, segmented tracks as well as detected and classified km-signs) are only available within our group (PFI). However, this data can also be of great value in other projects. Thus, we propose to process, unify and standardize this data in order to hand it over to the LocLab.

7.6.3 Further evaluation and testing

Prior to deployment the optical localisation method has to be further evaluated and tested. Additionally, the method has to be benchmarked and tested against the other investigated localisation methods (optical flow, GNSS, IMU, odometry, FOS). Therefore, we propose to generate additional data for a diverse set of railway sections and various conditions. To enable cross-modular evaluation and comparison, the additional data should be generated simultaneously with the data required for the other localisation methods.

7.6.4 Sensor fusion

Finally, a safe and reliable train localisation is only guaranteed using a combination of the different localisation approaches (optical localisation, optical flow, GNSS, IMU, odometry, FOS). Thus, sensor, e.g. through a Kalman-Filter, has to be investigated. Especially the combination of optical flow, IMU or odometry with optical localisation appears promising. Here, optical flow, IMU or odometry could be used to update the train position at a fast rate, while optical localisation can be used to avoid scale drift.

8 References

[1] Zwischenbericht Technologie PoC Lokalisierung, „Smartrail 4.0 Fachpublikationen,“ 18 January 2019. [Online]. Available: https://www.smartrail40.ch/service/download.asp?mem=0&path=\download\downloads\Integrierter_ZwischenberichtTechPocGLAT_v1.2_web.pdf. [Zugriff am 09 April 2020].

[2] Technologiebericht PoC Lokalisierung, “Smartrail 4.0 Fachpublikationen,” 29 January 2020. [Online]. Available: https://www.smartrail40.ch/service/download.asp?mem=0&path=\download\downloads\Anlage%20LCS_05%20-%20Technologiebericht-PoC_GLAT_v1.00.pdf. [Accessed 09 April 2020].

[3] Robotics Knowledgebase, „AprilTags,“ [Online]. Available: https://roboticsknowledgebase.com/wiki/sensing/apriltags/. [Zugriff am 03 April 2020].

[4] OpenCV, [Online]. Available: https://opencv.org/.

[5] IDS, uEye IDS-Camera Manual.

[6] „OpenCV Camera Calibration,“ [Online]. Available: https://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html.

[7] Mathworks, „Camera Calibration Toolbox for MATLAB,“ [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/.

[8] Website of the NMEA 0183 Standard, [Online]. Available: https://www.nmea.org/content/STANDARDS/NMEA_0183_Standard.

[9] P. G. Howard und . J. S. Vitter, „Fast and Efficient Lossless Image Compression,“ *Proceedings of the 1993 IEEE Data Compression Conference (DCC '93),* pp. 351-360, 1993.

[10] J. F. R. Herrera und V. G. Ruiz, „Golomb and Rice coding,“ 10 July 2017. [Online]. Available: https://w3.ual.es/~vruiz/Docencia/Apuntes/Coding/Text/03-symbol_encoding/09-Golomb_coding/index.html. [Zugriff am 03 April 2020].

[11] *JPEG File Interchange Format (JFIF). ecma-international.org,* 2009, Retrieved 15 June 2015..

[12] ISO/IEC 10918-5:2013, Information technology — Digital compression and coding of continuous-tone still images: JPEG File Interchange Format (JFIF) — Part 5.

[13] Definition of the OpenCV function Canny, [Online]. Available: https://docs.opencv.org/3.4.6/dd/d1a/group__imgproc__feature.html#ga04723e007ed888ddf11d9ba04e2232de.

[14] Definition of the OpenCV function HoughLinesP, [Online]. Available: https://docs.opencv.org/3.4.6/dd/d1a/group__imgproc__feature.html#ga8618180a5948286384e3b7ca02f6feeb.

[15] Definition of the OpenCV function adaptiveThreshold, [Online]. Available: https://docs.opencv.org/3.4.6/d7/d1b/group__imgproc__misc.html#ga72b913f352e4a1b1b397736707afcde3.

[16] Definition of the OpenCV class FastLineDetector, [Online]. Available: https://docs.opencv.org/3.4.6/df/d4c/classcv_1_1ximgproc_1_1FastLineDetector.html.

[17] J. H. Lee, S. Lee, G. Zhang, J. Lim und W. K. C. a. I. H. Suh, „Outdoor place recognition in urban environments using straight lines," in *IEEE International Conference on Robotics and Automation (ICRA), pages 5550-5557. IEEE*, 2014.

[18] Definition of the OpenCV function Template Matching, [Online]. Available: https://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template_matching/template_matching.html.

[19] Definition of the OpenCV function goodFeaturesToTrack, [Online]. Available: https://docs.opencv.org/3.4.6/dd/d1a/group__imgproc__feature.html#ga1d6bb77486c8f92d79c8793ad995d541.

[20] Definition of the OpenCV function calcOpticalFlowPyrLK, [Online]. Available: https://docs.opencv.org/3.4.6/dc/d6b/group__video__track.html#ga473e4b886d0bcc6b65831eb88ed93323.

[21] J.-Y. Bouguet, Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm., Intel Corporation, 5, 2001.

[22] Definition of the OpenCV function findEssentialMat, [Online]. Available: https://docs.opencv.org/3.4.6/d9/d0c/group__calib3d.html#ga13f7e34de8fa516a686a56af1196247f.

[23] Definition of the OpenCV function recoverPose, [Online]. Available: https://docs.opencv.org/3.4.6/d9/d0c/group__calib3d.html#gadb7d2dfcc184c1d2f496d8639f4371c0.

[24] „OpenStreetMap," [Online]. Available: https://www.openstreetmap.org. [Zugriff am 20 02 2020].

[25] Y. Bar-Shalom, X. Rong Li und T. Kirubarajan, Estimation with Applications to Tracking and Navigation, John Wiley & Sons, Inc., 2001.

[26] Machbarkeitsstudie für eine genaue, sichere Lokalisierung (Phase 0), [Online]. Available: https://smartrail40.ch/index.asp?inc=downloads.asp&typ=Nav2&cat=53.

[27] D. B. Spiegel, Qualifizierung sicherheitsrelevanter satellitenbasierter Ortungssysteme für den Bodenverkehr. Dissertation, 2018: Technische Universität Braunschweig, Institut für Verkehrssicherheit und Automatisierungstechnik, Braunschweig.

[28] D. Burschka and C. Robl, "Highly Accurate Video-Based Train Localization - replacing Balises with Natural Reference Points," in *The European Navigation Conference ENC 2020*, Dresden, Germany, May 11-14, 2020.

9 Glossary

See also link to glossary Polarion: https://trace.sbb.ch/polarion/#?shortcut=Glossar%20Deutsch

AprilTag	AprilTag is a visual fiducial system, useful for a wide variety of tasks including augmented reality, robotics, and camera calibration. (https://april.eecs.umich.edu)
CNN	Convolutional neural network
dB	Decibel: Decibel is an auxiliary unit of measurement to indicate the sound pressure.
DfA	Topology database (Datenbank der festen Anlagen)
DFT	Discrete Fourier Transform
DFZ	SBB diagnostic vehicle
ESF	Entropy Spectral Flatness
ETCS	European Train Control System, signalling and control component of the European Rail Traffic Management System
FFT	Fast Fourier Transform
FN	False Negative (Incorrectly identified as Positive)
FOS	Fiber Optic Sensing
FoV	Field of view
FP	False Positive (Incorrectly identified)
GAMAB	Globalement au moins aussi bon – Generally at least as good: A new system should be at least as safe or low-risk as any existing comparable system (cf. European railway standard EN 50126, 1997).
GGA	NMEA sentence containing time, position, and fix related data
GNSS	Global Navigation Satellite System
GPU	Graphics processing unit
GTG	GTG - GleisTopoGraphie
GUI	Graphical User Interface
IMU	Inertial Measurement Unit
IR	Infra-red radiation: electromagnetic radiation with wavelength from 700 nanometers to 1 millimeter
LiDAR	Light Detection and Ranging
MSE	Mean Squared Error
NIR	Near Infrared Radiation: electromagnetic radiation with wavelength from 700 to 1400 nanometers.
NMEA	Serial communications protocol that defines how data are transmitted in a sentence from one talker to multiple "listeners" at a time. https://www.nmea.org/content/STANDARDS/NMEA_0183_Standard
OBU	On Board Unit
PFI	Platform for Research and Innovation (Plattform für Forschung und Innovation)
PoC	Proof of Concept
PSD	Power Spectral Density

Qt	Application framework and GUI toolkit for cross-platform development of programs and graphical user interfaces.
RGB	Red green blue color image
RMC	NMEA sentence containing position, velocity, and time
SF	Spectral Flatness
SLAM	Simultaneous Localisation And Mapping
SNR	Signal to Noise Ratio
STFT	Short Time Fourier Transform
SWIR	Short wave infra-red
TP	True Positives (Correctly identified)
TPR	Train Position Report
Vanishing point	point on the image plane of a perspective drawing where the two-dimensional perspective projections (or drawings) of mutually parallel lines in three-dimensional space appear to converge.
YOLO	"You Only Look Once" convolutional neural network
ZDA	NMEA sentence containing UTC day, month, and year, and local time zone offset