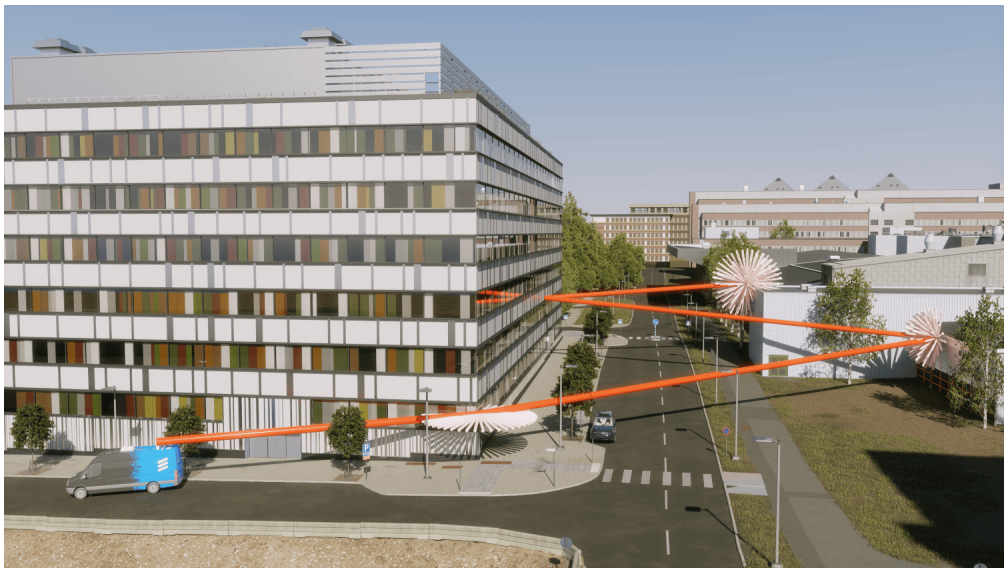
CHALMERS
UNIVERSITY OF TECHNOLOGY



Realistic Radio Propagation Modeling for a Digital Twin

Improvements with Enrichment of 3D Scenarios

Master's thesis in Physics

OSKAR MORE ARVIDSSON

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2023
www.chalmers.se

Master's thesis 2023

Realistic Radio Propagation Modeling for a Digital Twin

Improvements with Enrichment of 3D Scenarios

OSKAR MORE ARVIDSSON

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2023

Realistic Radio Propagation Modeling for a Digital Twin
Improvements with Enrichment of 3D Scenarios
OSKAR MORE ARVIDSSON

© OSKAR MORE ARVIDSSON, 2023.

Supervisors:
Martin Johansson, Ericsson Research
Gerhard Steinböck, Ericsson Research

Examiner:
Thomas Rylander, Chalmers University of Technology

Master's Thesis 2023
Department of Electrical Engineering

Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Visualization of a radio wave propagating in an urban environment.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2023

Realistic Radio Propagation Modeling for a Digital Twin
Improvements with Enrichment of 3D Scenarios
OSKAR MORE ARVIDSSON
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Radio waves at higher frequencies in the current and future generations of radio networks are more sensitive to details in the environment as they propagate. Previous measurements have shown that street furniture such as poles and trees can have a non-negligible effect on the characteristics of the radio propagation. Street poles can contribute significantly for around street corner situations to path gain, in particular at higher frequencies. Furthermore, the scattering contributes to the richness of the channel. For site specific modeling, ray tracing simulations are needed. By including street poles in ray tracing simulations in this work, more realistic simulation results were obtained. This was seen through changes in Doppler frequency and path gain. This thesis focuses on positioning the poles in the site specific modeling, enabling inclusion of them in the ray tracing simulations. To begin with, street view panorama images including the poles in the interesting area were extracted. With deep learning algorithms, object detection was performed through panoptic segmentation and range detection through monocular depth estimation on the extracted images. Given the direction and distance output from images extracted at multiple camera positions, street poles along a road are positioned through triangulation and clustering with a positioning error of 3.5 meters. This is comparable to related approaches in the field. The errors are mostly due to limiting GPS accuracy for camera positioning and limitations of detecting distant poles.

Keywords: radio propagation, ray tracing simulation, scattering models, poles, street view images, object detection, monocular depth estimation, geolocation

Acknowledgements

First, I wish to thank my supervisors Martin Johansson and Gerhard Steinböck at Ericsson for their engagement and excellent guidance. In addition, I wish to thank my examiner Thomas Rylander at Chalmers for his support and for together with my manager Henrik Sahlin at Ericsson giving me the opportunity to perform this thesis. Further, thanks to Remco Heijs at Ericsson for his support with the simulations, thanks to Lars Hammarstrand at Chalmers for his support with the positioning, and thanks to Georgios Spaias, Vasilis Naserentin and Anders Logg at DTCC for sharing their related work on environment recreation. Finally, thanks to my family, friends, and colleagues for their moral support and advice along the way.

Oskar More Arvidsson, Gothenburg, June 2023

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

ANN	Artificial Neural Network
API	Application Programming Interface
BS	Base Station
CNN	Convolutional Neural Network
CRS	Coordinate Reference System
CTF	Channel Transfer Function
EHF	Extremely High Frequency
EPSG	European Petroleum Survey Group
FOV	Field Of View
GSV	Google Street View
GPS	Global Positioning System
GPU	Graphics Processing Unit
IoU	Intersection over Union
LOS	Line Of Sight
MAE	Mean Absolute Error
MDE	Monocular Depth Estimation
MIMO	Multiple Input Multiple Output
mIoU	mean Intersection over Union
NLP	Natural Language Processing
PEC	Perfect Electric Conductor
RCS	Radar Cross Section
ReLU	Rectified Linear Unit
RGB	Red, Green, Blue
RMSE	Root Mean Square Error
PQ	Panoptic Quality
SHF	Super High Frequency
UE	User Equipment
USD	Universal Scene Description
UTD	Uniform Theory of Diffraction

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

Sets

TP	True positives
TN	True negatives
FP	False positives
FN	False negatives
D	Predicted depth values

Variables

w	Network weights
b	Network bias
$pred$	Predicted depth value
gt	Ground truth depth value
N_p	Number of predictions
N_m	Number of matched predictions
P	Power
G	Gain
PG	Path gain
d	Distance
f	Frequency
t	Time
λ	Wavelength
τ	Delay

ν	Doppler frequency shift
g	Field pattern
Ω	Path direction
α	Gain coefficient
$D(\nu)$	Doppler power spectrum
$h(\tau, t)$	Impulse response
$H(f, t)$	Frequency response
R	Reflection coefficient
σ	Radar cross section
N	Number of paths

Contents

List of Figures

List of Tables

1

Introduction

To develop future radio networks, realistic modeling of the networks is important to reach optimal design and performance. The digital modeling is performed by tracing radio waves between devices in the network, investigating how the rays interact with the environment. One possible way to make the simulations more realistic could be to enrich the simulation environments by including more details. Going beyond using outlines of buildings and streets, simulations in an urban environment could include street furniture like trees, lamp posts and cars from which the radio waves can bounce off.

1.1 Problem formulation

The purpose of this thesis is to investigate how enrichment of the 3D environment could make the ray tracing simulation of a radio network more realistic. This is studied by investigating how the electromagnetic characteristics of the radio network in the simulation may change with the inclusion of street furniture compared to the previous model. In order to obtain a sufficiently detailed environment, this thesis aims to detect, classify and position the street furniture from street view images. Using implemented electromagnetic models for the objects, the simulation results for the radio channel characteristics will be analyzed for indications of a more realistic digital twin.

1.2 Background

For new generations of radio networks, such as 5G and 6G, the requirements and design have become more complex [1]. An increasing population with an increasing number of connected devices demands wireless connection in an increasing area. With the development of ray tracing simulations and computational power, the possibility of simulating networks and designing them to achieve the best performance has improved [2]. By simulation and recreation of the physical product, one can analyze the performance more efficiently. This is the concept of a digital twin, and it has become a successful concept in topics ranging from product cycles to recreating city centers [3]. For the digital twin of a radio network, the electromagnetic characteristics should be as similar to the real network as possible. This of course

has to be with consideration to how many details actually are relevant seen to the overall performance.

One way in which this digital twin of the radio network could be improved to be more realistic is through enrichment of 3D scenarios. With enrichment of the scenarios, it is here meant to include street furniture [4] in addition to buildings and street outlines already included. Street furniture can be objects such as trees, lamp posts, street signs and cars to name a few. Relevant work done by Ericsson [5] and others [6, 7, 8] indicates that including street furniture in the ray tracing simulation environment can actually have a non-negligible effect on the radio propagation characteristics. More specifically, deviations between measurement and simulation are observed in path gain and in frequency shift through the spread in Doppler due to the moving receiver in their scenario. This may be due to the street furniture having an effect on the real radio channel characteristics of the measurements, but its effect is lacking in the ray tracing simulation.

To create a 3D environment as realistic as possible, it is important to recreate real scenarios including street furniture. Recreating specific scenarios is one approach, and using statistical rules applied to approximately include the correct amount of positioned street furniture is another. To make the specific recreation faster and easier, or at least improve the statistical rules to be more specific to a city or area, an automated workflow to detect street furniture can be applied. To find and position street furniture, street view images from urban environments can be used. Object detection and positioning algorithms can be performed [9, 10] with these images. A neural network can be trained on large sets of annotated image data including different object classes to handle the detection task. One example is the network in [11] that has been trained on the extensive Mapillary Vistas dataset including street furniture such as vegetation, lamp posts and cars as object classes [12]. The extracted positioning information about the details in the urban environment can then be merged into the scene description for an analysis of the impact on the ray tracing simulations.

1.3 Limitations

As stated in section 1.1, the aim of the thesis is to detect and position street furniture from images to enable enrichment of the 3D environment for ray tracing simulations. Especially, the work focuses on one kind of street furniture, namely poles of street signs and street lights. This is because poles are commonly encountered in urban scenarios, and the conductive material has a relatively clear influence on the radio waves compared to other materials such as wood. The clear vertical shape of the poles also gives the opportunity for more accurate detection and positioning, as well as more accurate electromagnetic modeling, as compared to trees. Compared to other street furniture such as cars, the poles are also static and not commonly changed. This is more suitable for recreating specific scenarios. Furthermore, it is especially the inclusion of poles that indicates non-negligible effects in [7, 8] and in measurements performed by Ericsson. The measurements are performed in Kista,

Stockholm, which for that reason will be the focused area in this thesis.

Despite the focus on poles, the aim is still to keep the positioning process as generic as possible in order to be useful for other street furniture as well and also suitable for other locations. The contribution of this thesis will be to combine and integrate the methods of automatic image analysis and radio propagation simulations. This can hopefully contribute to how digital twin generation could be more realistic by enrichment of 3D scenarios in a more general and automated way.

1.4 Ethics and sustainability

Regarding the street view images, consideration for privacy and integrity has been taken since the images used have blurred areas covering for instance faces and registration plates. The part of the thesis work including the image analysis and deep learning is based on existing imagery and open-source implementations. The choice of pre-trained models is a sustainable choice since training deep neural networks is energy consuming. Similarly, it is sustainable to use already existing measurement data for the radio propagation study.

1.5 Report structure

Following this thesis introduction, the theory chapter will introduce the topics of radio propagation and image analysis with machine learning on a basic level needed to follow the main part of the thesis. The upcoming method chapter describes the workflow including the creation of the enriched environment through image extraction, object detection and object positioning. With a similar order the result chapter presents and visualizes the step by step achievements. This is followed by a description of the ray tracing simulation and the results obtained with this enriched environment. Finally, the conclusion chapter sums up the main discussion points and contributions of this thesis.

2

Theory

This chapter presents the underlying theory behind the methods implemented in this thesis work. First, there is an introduction to the basic theory of radio propagation to set the scene for the digital twin of the radio network. Secondly, the general theory of neural networks for image analysis including object detection is covered which is the base for the creation of the enriched 3D environment.

2.1 Radio propagation

The concept of radio propagation modeling is to analyze how radio waves propagate or travel between points in space. Similar to light waves, radio waves are electromagnetic waves that can be characterized by frequency, amplitude, phase and polarization. These waves are attenuated or weakened with the traveling distance and can also undergo reflection, scattering and diffraction as they propagate. Analyzing the propagation of radio waves through measurements and simulations is important for the design of radio networks [13].

2.1.1 Radio network

In a radio network, the electromagnetic signals of the wireless communication are sent from central base stations (BS) strategically placed to reach a majority of the user equipment (UE). The BS usually has an antenna system mounted on a mast or building that communicates with a large number of UEs, such as mobile phones or other connected apparatus. The sending device is referred to as the transmitter and the target device as the receiver. The development of wireless telecommunication opened the door for vast opportunities and that a connection to a 4G network today can be reached in most areas. The increasing amount of connected devices and the increase in data traffic require better communications. In recent years, the development and implementation of the next generation, 5G, has spread in larger cities, and research on 6G networks proceeds.

The 5G network can handle an increased number of devices in crowded areas better than previous generations. One contribution to this is that new parts of the frequency spectrum are used for communication. The frequency of the signals in the 5G network can cover parts of the spectrum from just below 1 GHz, similar to 4G

networks, and up to around 50 GHz. The range of around 400-900 MHz is referred to as the low-band, 1.7-4.7 GHz as the mid-band and 24-47 GHz as the high-band. The mid-band is the most widely used, with around 3 GHz being the frequency implemented in most big city 5G networks. The frequency range 3-30 GHz is referred to as the super high frequency (SHF) or centimeter wave band associated with the wavelength, whereas 30-300 GHz is referred to as extremely high frequency (EHF) or millimeter wave (mmWave) [14].

Radio waves with a frequency higher than a few MHz follow the so-called line of sight (LOS) propagation [15]. This means the rays travel in direct paths. They may be diffracted, reflected or scattered on the way between the transmitter and receiver, but they will not bend and follow the contour of the Earth around the horizon like lower frequency waves.

Between the transmitter, which in an example could be a base station, and a receiver, which could be a mobile phone, there are so-called Fresnel zones. The Fresnel zones include possible LOS propagation paths. The first Fresnel zone includes the strongest signal. The radio wave propagation through space is in ray tracing modeled as a ray following a straight line from BS to UE as in Figure 2.1. The radio wave has additional characteristics such as amplitude and polarization that are taken into account.

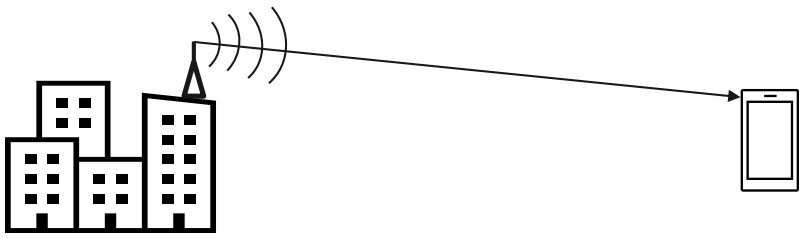


Figure 2.1: Visualization of the propagation of a radio wave modeled as a ray from the BS mounted on a building to the UE on the street.

For 5G radio networks, the base station antennas are phased array antennas. In combination with an antenna array for the user they form a multi-input multi-output (MIMO) antenna system [16]. The definition of a phased array antenna is that it is a combination of smaller regular antenna components, whose output phase can be shifted in order to form a beam of the signal to the desired direction and power through interference. This design enables a better directivity of the antenna, which means that more of the antenna output power is going to the desired direction towards the receiver. Thus, there is not only the BS position but also several parameters such as the directivity, the gain and the transmitted power that can be varied to optimize the network.

The power of the propagating radio wave between a transmitter and a receiver can be described with the radio equation, also known as Friis equation [15]. This equation is valid for free space and line of sight propagation for one radio wave, which is

assumed in an initial stage. The equation expresses the received power P_r as

$$P_r = P_t G_t G_r \left(\frac{\lambda_0}{4\pi d}\right)^2 \tag{2.1}$$

where P_t is the power fed to the transmitting antenna. G_r and G_t are the antenna gains of the receiving and transmitting antenna systems, respectively. The antenna gain describes the directivity and radiation efficiency of an antenna. An antenna gain of one would mean a theoretical isotropic antenna with identical characteristics in all directions. However, real antennas are not isotropic but instead more sensitive or output more power in a specific direction, and hence have a different gain. The distance between the antennas is denoted by d and the wavelength of the radio wave by λ. The wavelength is calculated as $\lambda_0 = c_0/f$ where c_0 is the speed of light in vacuum and f the frequency. Together, they compose the term $\left(\frac{\lambda}{4\pi d}\right)^2$ which describes the so-called free-space path loss [15].

Dividing the received power by the transmitted power gives the path gain

$$PG = P_r/P_t \ . \tag{2.2}$$

The path gain is often used as a measure of the performance of a radio network, since it relates the available power for a receiver to the transmitted power. Similarly, the path loss expressed as $1/PG$ can be used. Calculating the PG for several receiver positions would yield a coverage map relating to how good connection a UE would have at different positions in an environment.

2.1.2 Channel modeling

Characterizing a full radio channel is more complex than for one radio wave in LOS propagation. Channel modeling is needed for networks with antenna arrays. A radio channel including N radio wave paths can be characterized by the impulse response [13] similar to

$$h(\tau, t) = \sum_{n=1}^{N} g_{Tx}(\mathbf{\Omega}_{Tx,n}) g_{Rx}(\mathbf{\Omega}_{Rx,n}) \alpha_n e^{-j2\pi f_c \tau_n} \delta(t - \tau_n) e^{j2\pi \nu_{Rx,n} t} \ . \tag{2.3}$$

Here, g_{Tx} and g_{Rx} are the field patterns describing the directivity of the transmitting antenna Tx and receiving antenna Rx, respectively. These field patterns are functions of the respective path directions $\mathbf{\Omega}_{Tx,n} = [\varphi_{Tx,n}, \theta_{Tx,n}]$ and $\mathbf{\Omega}_{Rx,n} = [\varphi_{Rx,n}, \theta_{Rx,n}]$ of each path n. Given a spherical and horizontal coordinate system, φ is the azimuth angle and θ the elevation angle from the vertical axis as shown in Figure 2.2.

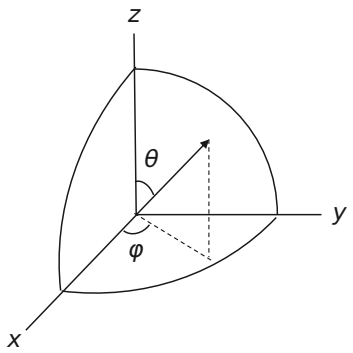


Figure 2.2: Spherical coordinate system with the angles φ and θ describing the path direction marked.

The individual gain for each path is denoted α_n and is dependent on the type of interactions such as line of sight, reflection, diffraction and scattering it passes. The expression for α_n is further described in subsection 2.1.3 below. Each path also exhibits a certain phase rotation, $e^{-j2\pi f_c \tau_n}$, dependent on the paths delay τ_n and carrier frequency f_c. The delay is due to the distance traveled by the radio wave, and causes the shift represented in Equation 2.3 by the Dirac delta function as $\delta(t - \tau_n)$.

Finally, there is another phase shift due to the mobility of the receiver. This is due to the Doppler effect [17] that results in a Doppler frequency shift

$$\nu_{Rx,n} = \frac{\boldsymbol{r}_{Rx,n}^T(\boldsymbol{\Omega}_{Rx,n})\boldsymbol{v}}{\lambda_0} \ . \tag{2.4}$$

In this expression, the velocity vector of the receiver is dependent on the angles of movement as $\boldsymbol{v} = v[\sin(\theta_v)\cos(\phi_v), \sin(\theta_v)\sin(\phi_v), \cos(\theta_v)]$ where v is the absolute velocity in the direction of the movement. The path's directional vector is described by $\boldsymbol{r}_{Rx,n} = [\sin(\theta_{Rx,n})\cos(\phi_{Rx,n}), \sin(\theta_{Rx,n})\sin(\phi_{Rx,n}), \cos(\theta_{Rx,n})]$. In Equation 2.4, $\boldsymbol{r}_{Rx,n}^T$ is the transposed vector. These vectors are given in Cartesian coordinates.

The Doppler effect for electromagnetic radio waves works similarly as for sound waves. With a moving receiver relative the transmitter as one example, the frequency will decrease for a receding receiver and increase for an approaching receiver. The effect is further described in [17].

2.1.3 Interactions

Radio waves of the new generation networks in the SHF and EHF range are extra sensitive for interactions with objects due to the short wavelength. With advancement of computational performance through graphics processing units (GPUs), the possibility to model these detailed interactions has improved. This is why ray tracing simulations are more widely used as one essential part of analyzing radio networks. With the higher sensitivity, it becomes more complex to model and optimize the characteristics through the placement of the antennas and the choice of frequency [15].

For a ray tracing simulation in an urban environment with many obstacles, the interactions of the rays with buildings, streets and street furniture have to be modeled. Without interactions and only line of sight propagation, as described above in Equation 2.1, the gain for each path is described as

$$\alpha_{LOS} = \frac{\lambda_0}{4\pi d} , \tag{2.5}$$

in other words the free-space path loss.

For a plane wave incident on a planar material interface, Snell's laws describe how the angle of incidence is related to the angle of reflection and transmission, where the angle of transmission is dependent on the materials. Furthermore, the Fresnel coefficients describe magnitude and phase of the reflected and transmitted wave in relation to the incident wave, where these coefficients are dependent on both the material and the polarization [18]. On a smooth surface of a perfect electric conducting (PEC) material, the radio wave will undergo specular reflection and the reflection coefficient is $R = -1$. The gain of the path is described with the coefficient

$$\alpha_R = R\frac{\lambda_0}{4\pi(d_1 + d_2)} , \tag{2.6}$$

where d_1 is the distance from the transmitting antenna to the object and d_2 is the distance from the object to the receiver.

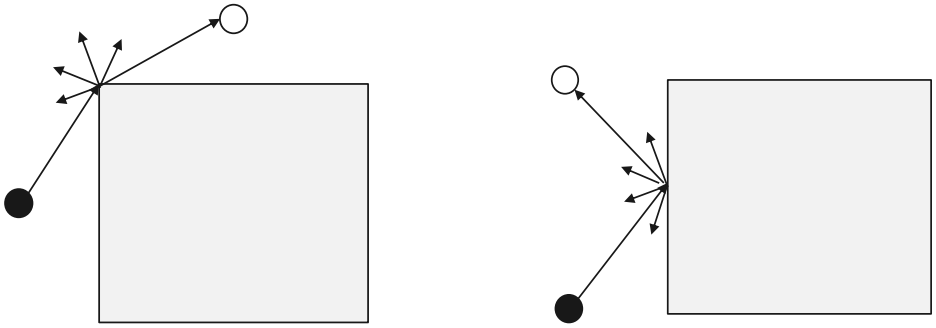
However, with real world objects such as buildings this is seldom the case since the building facades are rough and the material is not always PEC. For non-PEC materials $|R| < 1$, which can be obtained for various material types, thicknesses and incoming angles according to [18]. For rough materials, one common model is the Lambertian diffuse scattering model [19]. The model approximates the surface roughness and the backscattering due to small-scale geometric variations over the surfaces. The diffuse scattering spreads out the rays in a hemisphere which is visualized in Figure 2.3b. Also the diffraction, which is the interaction of rays on and around edges and corners of buildings, is visualized in Figure 2.3. A common model of diffraction is the uniform theory of diffraction (UTD) [20].

In the work related to this thesis, the interactions with street furniture are treated in a special way. To model these interactions, two paths of line of sight propagation before and after the obstacle are considered. The influence of the obstacle is added through a calculated radar cross section (RCS) σ. The calculation of the RCS follows the radar equation [15]. The individual gain of the paths is obtained as

$$\alpha_{Scat} = \frac{\lambda_0\sqrt{\sigma}}{(4\pi)^{3/2}d_1d_2} \tag{2.7}$$

where d_1 and d_2 once again are the respective distances from the transmitter to the obstacle and from the obstacle to the receiver. The radar cross section will depend on the incoming and outgoing angles of the paths [8].

The street furniture in focus for this thesis is poles of street lights and signs. This is in part because in recent work such as [7, 8], it was shown that poles can contribute significantly to path gain (PG). For scattering on a pole, the radar cross



(a) Diffraction. **(b)** Diffuse scattering.

Figure 2.3: A ray propagating from a transmitter in black to a receiver in white, it interacts with the obstacle in grey through diffraction in Figure 2.3a and diffuse scattering in Figure 2.3b.

section can be modeled analytically which has been used as a base for the current implementation. In [21] the radar cross section is provided for the far field. Far field means that at a great distance from the pole, the electromagnetic field is decreasing inversely proportional to the increasing distance. This is given a signal that is not incoming from the top of the pole. Both polarizations are handled separately. To fulfill this assumption also on shorter distances for typical frequencies, the pole has been divided into smaller sections. As the specific electromagnetic modeling is not part of this thesis, the reader is referred to the provided references [7, 8] for more details.

Other types of street furniture can also be included in radio propagation modeling. Trees and vegetation are interesting scattering objects. In a simplified model, they can be seen as a medium other than free space through which there is line of sight propagation. Then another term on top of the free space loss is multiplied to Equation 2.1 to account for the stronger attenuation through the vegetation. More complex scattering models for trees could also be analyzed and possibly implemented similar to as described in [22], but the complexity of the tree shape and medium makes it even more difficult and computationally costly than for poles.

2.2 Image analysis

In this section, the theory of analyzing image material to find the street furniture interesting for the ray tracing simulations is covered. Extracting information from images may be seen as a fairly simple problem, since a person from looking at a picture quite easily can get a perception of what it depicts and where in the scene the features are. With the rise of computing power and the field of deep learning, the analysis can now be automated. In this way, detecting objects and recreating scenes from imagery can be possible with less manual work, but the problem of getting there is not as simple. The theory of general neural networks is covered, followed by the more complex networks for object detection implemented in this thesis. Finally, an introduction to neural networks for distance estimation in images

will be presented.

2.2.1 Neural networks

Inspired by the function of human brains, artificial neural networks (ANN) or simply neural networks are built to recreate how the brain can learn from multiple inputs throughout life to recognize patterns, situations and objects. ANNs are usually built with several layers of neurons between which signals, represented by numbers, can be processed. The signals are multiplied with individual weights depending on between which neurons they pass. Figure 2.4 shows the most simple ANN architecture, with neurons divided into input and output layers with a hidden layer between them. From each neuron, the signal is processed with a weight contributing to the value of the neurons in the next layer. In supervised learning, the network is trained with a large amount of input data for which the classification is known. In the training process, the ANN adjusts the weights between neurons so that it can classify new test data based on what has been learned previously [23].

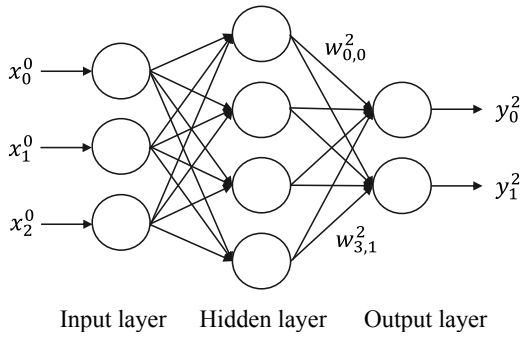


Figure 2.4: Outline of a basic ANN with neurons in input, hidden and output layers between which the input signals x are processed with weights w to output signals y.

For the most simple neural network architecture shown in Figure 2.4, the forward propagation of the signals through the network is shown in Equation 2.8. There, the output signal y_m^l for the output layer numbered l and neuron number m is calculated. The expression includes a sum of the values from all neurons x_n^{l-1} in the layer before, where each value has been multiplied with an individual weight $w_{n,m}^l$ between neuron n in layer $l-1$ and neuron m in layer l. Additionally, a bias term b_m^l is added.

$$y_m^l = \sum_n x_n^{l-1} w_{n,m}^l + b_m^l \tag{2.8}$$

To set the scene, one commonly used example of the application of basic ANNs is the recognition and classification of handwritten digits in the MNIST dataset [24]. Then the input values to the neural networks input layer are the values of the different pixels, projected on a one dimensional vector, and the output after processing through the network is an integer.

2.2.2 Convolutional neural networks

Convolutional neural networks (CNN) are a type of ANNs that were developed for image analysis. In comparison to the most basic artificial neural networks, CNNs can better handle multidimensional input such as the different pixel values of an image. For the MNIST dataset, CNNs significantly improve the classification accuracy compared to basic ANNs [25]. In scaling the application to more complex images and detection tasks a basic CNN will be necessary. The special design of the CNNs makes it possible to reduce the overall number of weights or parameters in the network, hence reducing both computational time and the risk of overfitting. Overfitting means that there are so many parameters to adjust that the overall result will fail to solve the general task of classifying test data.

The general architecture of a CNN is sketched in Figure 2.5. The input image is split into different depth layers, such as the three color layers red, blue and green (RGB) for images with color. For a grayscale image, one depth layer would be enough. The image is processed in convolutional layers, pooling layers and fully connected layers, respectively, before the signal is flattened and outputted as a label of what the image is picturing [23]. The smaller square visualizes the flow of a couple of pixels through the network.

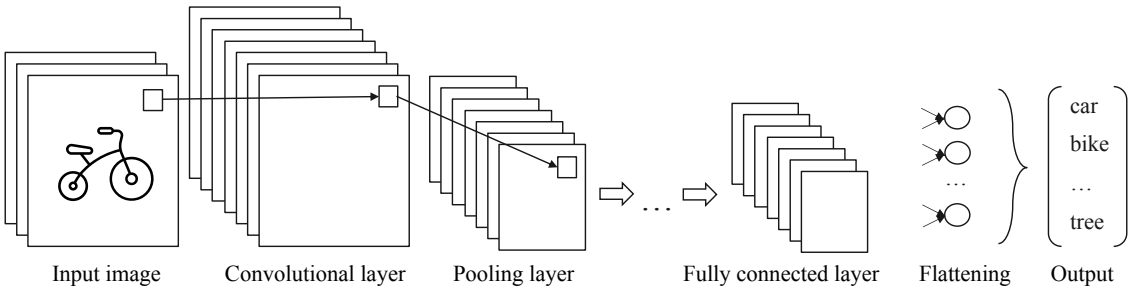


Figure 2.5: Outline of a CNN with an input image, a convolutional layer, a pooling layer, two fully connected layers and a flattened output with related classification labels. The stacked rectangles indicate different kernels and the small rectangle the flow for a set of pixels.

In the convolutional layers different convolutional kernels, as sketched in the depth dimension, are applied to parts of the input matrices. These are typically of dimension 3x3 or 5x5 including weights and are applied with regular scalar product, preserving the height and width dimension. The scalar multiplication is followed by an elementwise activation function in the form of the rectified linear unit (ReLu), returning $\max(0, x)$ where x is the element. This activation function helps reduce overfitting by setting negative values to zero. There are different kernels learned to classify low level details such as edges or general shapes in the image.

The pooling layer then reduces the dimensionality with downsampling, where neighboring pixel values are associated together resulting in one common value. The best performance achieved is with maximum pooling, extracting the maximum value of the region, since it suppresses noise from earlier values set to zero by the ReLu. An-

other option could be average pooling taking the average of included values, which as described would be more sensitive to noise. Between the convolutional layer and the pooling layer in Figure 2.5, or after the pooling layer, another convolutional layer can be applied to repeat the process. In the new layer, the kernels can be trained to recognize more high level and feature specific characteristics such as the wheel of a bike or the wing of a bird. The inclusion of additional convolutional layers and even pooling layers can be repeated multiple times.

After the last pooling layer, what would be classified as enough details of the image have been detected. In addition, the dimensionality of the neurons has decreased to a manageable size such that two fully connected layers, with functionality similar to the most basic ANNs, can be applied. Between these another ReLu activation function can be applied to further increase the performance and reduce the noise of a blurry picture for instance. Finally, by flattening the signals, one-dimensional output referring to the classification label of the image can be delivered. For instance, the image in Figure 2.5 can be labeled to include a car or a bike.

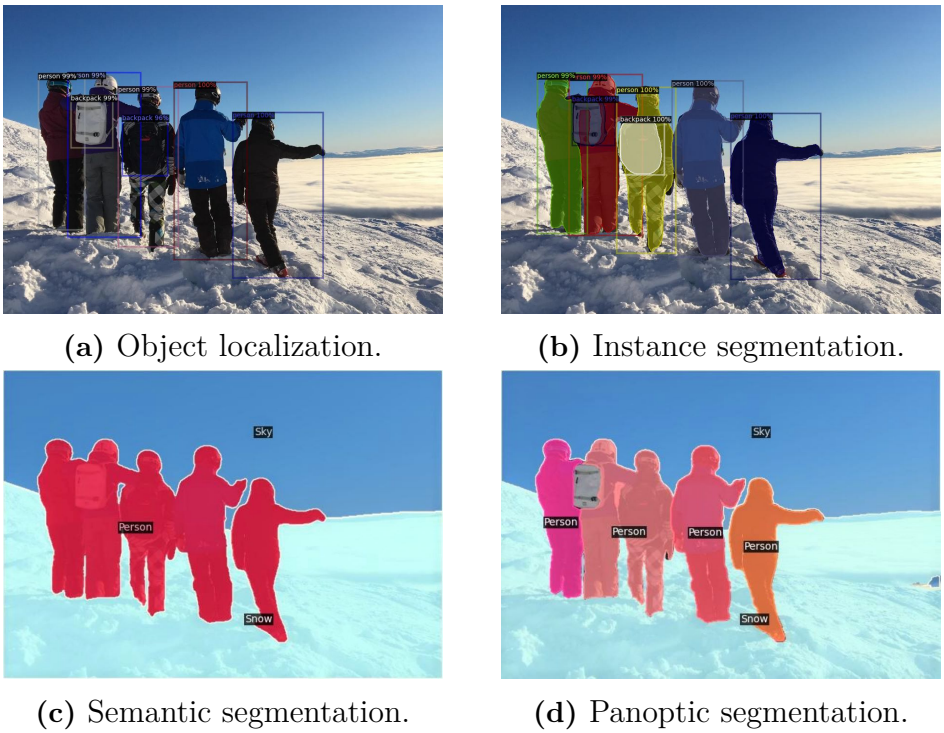
For a neural network to be ready to classify images as explained, it first has to be trained on a vast amount of data. The increasing size and variety of datasets have been one major reason for the development of the image analysis field. The datasets include annotated images, which means that the images are associated with a label of what it depicts. The variety of annotation labels, or object classes, determines what objects the network can predict in the test images. Two of the most widely used datasets are Microsofts Common Objects in Context (COCO) [26] and ImageNet [27], both with hundreds of object classes.

There is a variety of options to construct CNNs with different numbers of layers and different kernels. One of the best performing and most widely used architectures is ResNet, short for residual net, developed by Microsoft Research and setting records on both the image classification datasets COCO and ImageNet by the time of development [28]. The inclusion of residual learning, where the signal paths include skips over layers, reduces the impact of the vanishing gradient problem [23]. This problem means that occasionally small changes in the network parameters can make the optimization less accurate. The ResNet50 for instance is comprised of 50 layers, including five convolutional layers with combinations of 7x7, 3x3 and 1x1 convolutional kernels, and is commonly used as a backbone, where other algorithms have used this one as a first step and baseline to build upon.

2.2.3 Panoptic segmentation

The convolutional neural networks covered in subsection 2.2.2 can manage the image classification task. This means that given an image, the output will be the network's idea of what it represents. For some applications this is enough, but there are further steps in the development of neural networks for image analysis. A first step would be to not just classify, but also localize the object detected by creating a bounding box around it. This could be approximated by the ResNet from the resulting labels [28], but to include it in the training process the regional convolutional network

(R-CNN) [29] was developed. The R-CNN iteratively increases the analyzed region of the image to better define what part of the image that is relevant to analyze for the object in question, called the region of interest (RoI).



(a) Object localization. **(b)** Instance segmentation.

(c) Semantic segmentation. **(d)** Panoptic segmentation.

Figure 2.6: Visualization of the difference between object detection in the form of object localization, instance segmentation, semantic segmentation and panoptic segmentation.

Similar to investigating the regions of interest, the Mask R-CNN predicts a segmentation mask for the object in addition to the bounding box [30]. The segmentation mask is a coupled group of pixels that are all part of the object. The segmentation masks can be formed in different ways, where separating specified individual objects with separate masks is called instance segmentation. If individual objects of the same class are grouped together in the same mask, which is common with object classes that can be difficult to differentiate, it is called semantic segmentation. The difference between these segmentation methods is visualized in Figure 2.6. This version of semantic segmentation also includes classification of background pixels in the image. Combining classification of all pixels in the image with the separation of as many object masks as possible, similar to instance segmentation, is referred to as panoptic segmentation.

In panoptic segmentation, the object classes are often divided into ”things” and ”stuff”. Things like a person or car are objects that distinctly can be separated into individuals and that are countable. Stuff such as buildings, vegetation and roads are classified as regions and assumed uncountable. One of the most accurate and qualitative panoptic segmentation algorithms is developed by FaceBook AI

Research and named Mask2Former [11]. As with the previous methods, it builds upon a backbone pre-trained on the ImageNet dataset. It can be chosen to be a standard ResNet backbone, but a so-called Swin transformer backbone can enhance the performance even more [31]. Swin stands for shifted windows, which means it can adjust the size and placement of regional windows where the detection is performed. The transformer concept is inspired from natural language processing (NLP) where the language is broken down and analyzed from learned features. In a similar manner, the transformer for image analysis performs the detection with help from learned queries and features for the detection of object classes. In this way, the search is based on the known connection between queries and features rather than upsampling from pixel level to something similar to the object class.

The Mask2Former algorithm then combines feature pyramid extraction in a pixel decoder and a transformer decoder for the most accurate pixel classification throughout the picture. The pixel and transformer decoder are both specialized in detecting high resolution and feature specific details. They can in this architecture work together by complementing each other for a better result. The pixel decoder gradually covers more detailed areas and objects with a multi-scale approach on different resolutions. The transformer decoder is applying masked attention, which can take the multi-scale detection and complement with detection from learned features such that the extraction of the high resolution features is improved.

For panoptic segmentation, the most common evaluation metric is the panoptic quality (PQ) [32]. It can be seen as a metric on how well the predicted segmentation masks match the ground truth masks of the image. It is calculated as

$$PQ = \frac{\sum_{(p,q)\in TP} IoU(p,q)}{TP + \frac{1}{2}|FP| + \frac{1}{2}|FN|} \tag{2.9}$$

where TP is the collection of prediction masks p and ground truth masks q that match. FP is the collection of predicted masks not matched and FN is the collection of the remaining unmatched ground truth masks. Intersection over union (IoU) is a metric to show the overlap between matching prediction and ground truth masks formulated as

$$IoU(p,q) = \frac{p \cap q}{p \cup q}\,. \tag{2.10}$$

As an additional measure of the quality of the segmentation the mean IoU (mIoU) can be used as well, which is as the name describes a mean taken of the IoU values for all masks in the set TP, the predicted masks with matching ground truth masks.

A widely used dataset with segmentation masks is Mapillary Vistas dataset[12]. It includes street view images and is widely used for training algorithms for automatic driving. This dataset includes 25 000 high resolution images from cities covering 6 continents, with a variety of weather and camera settings. Furthermore, the images have been manually annotated with 124 semantic object categories of which 100 are instance specific.

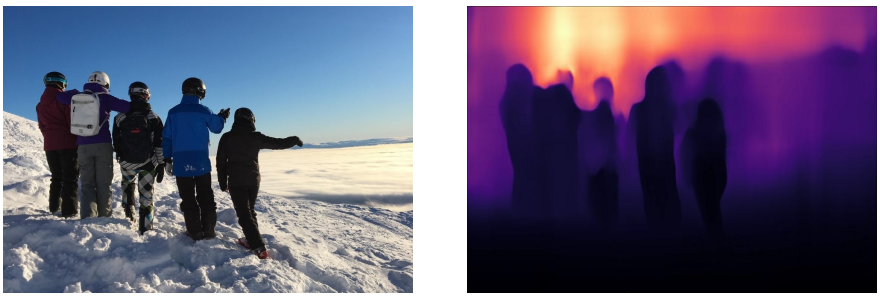
The use of the dataset from Mapillary for training and thus inclusion of street poles as an object class is the main reason for the choice of the Mask2Former algorithm.

In addition, it has the competitive panoptic quality of 45.5 and a mIoU of 60.8. Furthermore, it builds on the previously developed Facebook AI library `detectron2` [33] implemented with `PyTorch` which makes it relatively handy to implement.

2.2.4 Depth estimation

Associated with the field of object detection is range detection. For an autonomous car for instance it is not enough to know that there is another car in the vicinity and in which direction it is located. It is also crucial to know how far away it is. For range estimation there are three different sensors that can be used. Besides image based, these include radio detection and ranging (radar) and light detection and ranging (lidar). Radar is based on reflection of radio waves and lidar on reflection of light waves. However, the most accessible sensor is the camera. The signals from all these type of sensors can also be used together [34] to improve performance.

Only using the camera input could lead to estimations with relative big errors and narrow user applications compared to the other sensors. However, the range accuracy using images have improved. Predicting the range or depth in a single image, called monocular depth estimation (MDE), is a growing field with the development of deep learning [35]. Neural networks can be trained with the increasing number of image datasets with to predict relative depth in test images. This can be done in an self-supervised manner, learning depth estimates from stereo images from different directions and even complemented with video material, or with supervised learning and ground truth data. The trained algorithm can predict a depth map including values for each pixel in the image, commonly visualized on a colored map as in Figure 2.7b where brighter color indicates a longer distance.



(a) Original image. **(b)** Monocular depth estimation.

Figure 2.7: Visualization of a distance scale in Figure 2.7b from monocular depth estimation applied on the image in Figure 2.7a. Brighter color indicates longer distance.

The colored scale in Figure 2.7b enables a perception of the relative distances in an image. For instance, it can be seen that the ground covered in snow and the persons standing on it are closer to the camera than the sky. In order to relate the predicted depth scale to metric absolute distances between the camera and object, one has to take into account the characteristics of the camera such as focal length and the size of the image [36].

One of the best performing algorithms is named Monodepth2. The training dataset includes imagery captured with cameras of different focal length and thus field of view (FOV) leading to an algorithm with a general application. The general application of the Monodepth2 algorithm makes it possible to use for other image and camera types that are not necessarily included in the training dataset. As proposed in [36] and also implemented further in [9], the depth map can be scaled for application on a specific image type. This approach calibrates the scale from knowledge of the true data in a test set of images for a later general application.

Monodepth2 is a combination of a depth and pose decoder [36]. The depth decoder is a type of CNN called U-Net, which is an optimized version of a fully connected CNN to analyze the most detailed features of an image. The pose decoder is a slightly modified ResNet backbone as described in subsection 2.2.2, pre-trained on the ImageNet dataset. It is modified such that it from two different image frames can predict a relative pose of an object. Predictions from the depth decoder are in training compared to the pose prediction in a loss function to be minimized. This is performed on pixel level followed by upsampling to bigger regions for consistency over the image. The algorithm training is self-supervised. There is no prior annotated knowledge of the ground truth data, instead the self-supervised approach uses an automatic approach to approximate the truth from the image poses. The dataset used consists of stereo imagery, with different views on the same object, and monocular video material. The images depict street view scenarios.

The performance of depth prediction algorithms for MDE is typically evaluated on the KITTI dataset [37]. It is an image and distance dataset in which true distance values are provided from the collection with lidar sensors simultaneously with the image capturing. Different evaluation metrics can be used, where one of the most common ones is the root mean square error (RMSE) which is evaluated as

$$RMSE = \sqrt{\frac{1}{|D|} \sum_{pred \in D} ||gt - pred||^2} \tag{2.11}$$

where D is the set of all predicted depth values $pred$ for a single image, each of them compared to the ground truth gt. Monodepth2 receives one of the best root mean square errors (RMSE) of 4.63 meters [36].

3

Methods

In this chapter, the methods for the main track of the thesis workflow are described. First, it will be described how the 3D environment is created. The focus is then on the enrichment of the 3D environment. This includes the image extraction and the image analysis for which the theory has been covered, followed by the positioning of the objects in the environment.

3.1 Creating the 3D environment

In order to provide a digital twin of the radio network that is as realistic as possible, the 3D environment in which the ray tracing simulations are performed also has to be realistic. One of the areas in which measurements and simulations are performed by Ericsson is in Kista, Stockholm as introduced in section 1.3. To build up the 3D environment, terrain data is first fetched for the desired area of approximately 2000 times 1800 meters. This includes a height over sea level profile with a resolution of approximately one meter. In addition, so-called shapefiles including the outlines of buildings and street networks in vector format are fetched for the area. The terrain information and the shape outlines are then merged so that they overlap in a tool called CityEngine used for building 3D environments.

From the CityEngine tool, the model is exported as a universal scene description (USD) developed by Pixar. This USD format saves computer graphics data in three dimensions and is optimal for including different characteristics of an environment. USD files are commonly used for representing 3D environments in ray tracing tools. USD file data can be visualized in a variety of tools, such as Omniverse Create from NVIDIA. Omniverse Create can both build and use 3D environments for visual effects in an extensive way, where the visual effects of for instance windows can be included and different weather conditions can be applied. Also, the resulting rays from the ray tracing simulations, once performed, can be included for visualization in the environment of Omniverse Create.

3.2 Enriching the 3D environment

When enriching the 3D environment, proper image material of the area including the street furniture has to be extracted. This process is described in this section,

followed by how the images are processed with object detection to position the street furniture in the environment. An overview of the enrichment process can be seen in the flowchart in Figure 3.1. The flowchart includes the image extraction, the algorithms used with some of the processing steps implemented in this thesis as well as the two positioning methods with belonging clustering. This is an initial overview, where more detailed descriptions follow in this section. Furthermore, the poles are merged into the synthetic 3D environment where panoramas also can be extracted for an analysis of different error sources that are not present in a synthetic environment.

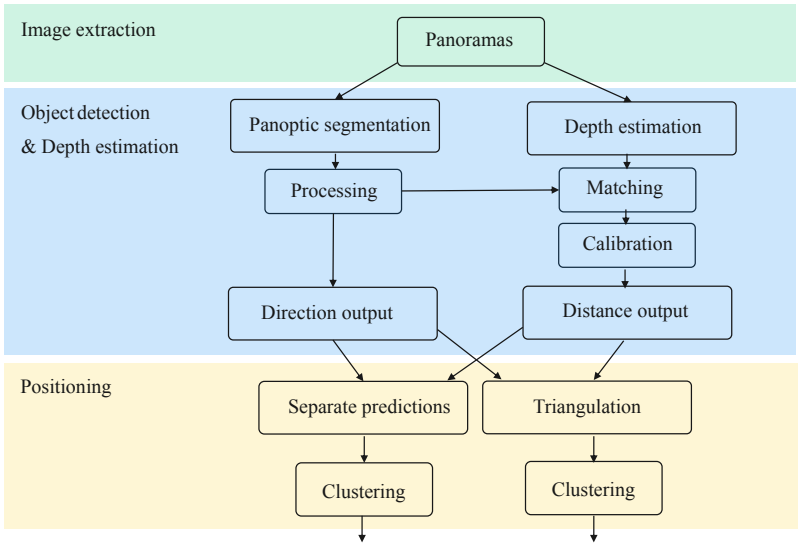


Figure 3.1: An overview of the process for enriching the 3D environment. It includes image extraction, object detection and positioning.

3.2.1 Image extraction

To begin with when enriching the environment, suitable image material is extracted from the area of interest. In this thesis, street view images from Google Street View (GSV) have been used since they cover many cities and also can deliver great image quality [38]. Other providers of street view images such as Mapillary [12] offered a limiting quality since they are created by private contributors to a greater extent. As a compliment, aerial or satellite images are extracted from Google Earth for an overview of the area [39].

For as good coverage of the environment as possible, panorama images with a field of view (FOV) of 360 degrees are extracted. The GSV application programming interface (API) can provide the available camera positions. Smaller tiles of the view are extracted and stitched together to form the full panorama. In total 16 tiles in the horizontal direction and 9 tiles in the vertical direction are stitched, with deliberate consideration such that the distortion is minimized in the panoramas [40]. The extracted images are after they have been stitched scaled down to the resolution 1024 x 320 from 6656 x 3328. This is performed with the interpolation method `INTER_AREA` from the Python package `OpenCV` [41] which interpolates the

pixel values depending on the area relation. The low resolution is chosen due to the specific requirements of the depth estimation algorithm Monodepth2 which requires specific resolutions.

Several panorama images along a street are extracted to populate a whole street scenario with street furniture in this thesis. These are extracted with an approximate distance of 10 meters between them, which is the shortest distance available and is chosen to maximize the number of images featuring each object. The GSV car is supposed to have a maximum speed of 45 km/h and capture at least one image every third meter. Filtering by Google based on image quality has then led to the publication of images approximately every 10 meters [42].

The Google API works such that it provides the closest available camera location given an approximated position. To extract the camera positions and panoramas in a certain area for this thesis, it was chosen such that the extraction is started with an approximate location in latitude and longitude coordinates at the beginning of a street segment. In addition, the approximate heading of the street is given. The heading is given in degrees in the range $[0, 360)$ similar to a compass, where 0 is North, 90 East, 180 South and 270 West. The first camera position is extracted from the API and is the closest to the initial guess. The guess for the second camera location will be at a distance of 10 meters in the approximated heading of the street. The camera location closest to this second guess will then also be saved. For the upcoming camera locations, the heading of the guess will be changed such that it follows the direction of the two previously saved camera locations.

In order to make calculations like these with the camera locations, the usual GPS coordinates given as spherical coordinates with latitude and longitude according to the World Geodetic System1984 (WGS84) standard are projected to a coordinate reference system (CRS) with Cartesian coordinates. There are many different CRS, each one has different coverage depending on the region on Earth. Regions can overlap and sometimes different CRS can be used in the same region, which makes it important to use the same transformation for all different coordinates included in the same project. For the Stockholm example above, the CRS with European Petroleum Survey Group (EPSG) code 32633 is used. This was chosen since there was already a shapefile provided for the area with this code. A shapefile is a file format saving geospatial vector data such as building polygons or road lines according to one of these reference systems. Given this reference system, a point close to the middle of the area of interest was chosen as the new center for the coordinates for easier visualization.

3.2.2 Object detection

Once the images containing the street furniture have been extracted along the road, the details are extracted with the object detection algorithm. A state of the art neural network for panoptic segmentation described in subsection 2.2.1 is used. The Mask2Former network used has been pre-trained on a big dataset from Mapillary [12], including many street view images from various parts of the world and with

various labels. Among the labels for things, that can be detected with separate masks in the image, there are the poles of street lights and street signs that are interesting for this thesis. The inclusion of poles as an object class in the dataset that the algorithm has been pre-trained on, in combination with the competitive performance, are the main reasons for the choice of Mask2Former. The evaluation of the model is performed with an external NVIDIA GeForce RTX 3090.

3.2.2.1 Mask processing

The output of the object detection algorithm is a matrix with the same dimensions as the pixel width and height of the input panorama image, in this case 1024 x 320. Every pixel detected as part of an object class is assigned a category identification number, relating to each specific car or in this case most importantly each pole. In a few cases, two poles are detected and classified with the same number as if they were one common pole. Because of the vertical nature of the poles, and the clear horizontal separation between them, the sorting of which pixels are part of which poles is possible. To limit the effect of false detections and poles too far away, a minimum number of pixels is set to 30 for a separated mask to be classified as a pole. Visual inspection indicated that the detection masks with fewer pixels than around this threshold were mostly false detections, poles far away or even small parts of the same pole where for instance the top of the pole is bent to hold up a lamp.

3.2.2.2 Direction estimate

To position the street furniture from the object detection, a horizontal direction from the camera position to the object is estimated. Since the field of view of the image is 360 degrees, the position of the pole mask relative to the width of the matrix can give the approximate angle of the pole [9]. Because of the vertical nature and relatively uniform thickness of the poles, and also in order to account for partial occlusion of the pole, the most vertical mask position in the image is used. With this it is meant that the number of pixels in the mask is projected on the horizontal axis, and the position with the highest number of pixels is chosen.

For this angle to make any sense for positioning it has to be related to a common reference. This reference is chosen as North, and in each panorama it is approximated from the knowledge that the exact middle of the panorama is the driving direction of the car. The driving direction is approximated from the direction to the next camera location for the panorama extractions as visualized in Figure 4.1. Since the CRS is in x- and y- coordinates, where the y-axis is pointing toward the North, this can be related to the direction of the North in the image. An alternative method to deciding the direction of North in an image would be to use image matching through `OpenCV` [41] with a smaller image extracted from GSV as well but with a previously known heading [40]. Despite the promising applications to panoramas in a few test cases, this method showed not to be accurate enough for the application in this work.

3.2.3 Depth estimation

The approach applied in this thesis for positioning the street furniture from object detection is in combination with range estimation. This is done with MDE as described in subsection 2.2.4, more specifically with the deep learning algorithm Monodepth2 [36]. This algorithm provides the metric distances estimation to each pixel of an image.

3.2.3.1 Distance estimate

A distance estimate for a pole can then be calculated from the depth values of the pixels associated to each pole mask from the panoptic segmentation. In the similar positioning approach with segmentation masks and depth estimations in [9], a single distance estimate for each mask was calculated as a trimmed mean of all the mask pixels. In [9], it was motivated to exclude the top and bottom 10 % of the pixel depth values, respectively, due to the difficulty of the algorithm to predict depth around the edges of the object. In [9], trees were detected and positioned. From an analysis of the general distribution of depth values for the pixels of a pole mask in our implementation, there was a tendency to predict a too long distance for many of the pixels. This was especially the case for the tops of the poles, probably since they are relatively thin. The final distance estimate for each pole was therefore calculated as a mean of the closest half of pixel values. As stated, this was done in order to account for the tops of poles that in this thesis work often were not detected as clear as the bottom part, and that the edges of the poles were not so clear causing predictions farther away.

Combining the known direction and distance estimates of the detected pole, a position can be predicted. Given the camera position where the panorama was extracted, the position prediction is at the estimated distance in the estimated direction. To position poles along a road segment, predictions from multiple panoramas are combined. With a combination of several camera positions as prediction sources, there might be multiple predictions associated with the same pole but detected from a different views. In subsection 3.2.4 the algorithms to combine these predicted positions are discussed.

3.2.3.2 Calibration

Note that the multiple predictions are used in a calibration procedure as in [9]. All the predictions are used to adjust the depth scale with a factor to match the focal length of the camera for the panoramic images from GSV. This factor was obtained as 0.25 for GSV images after comparison with real truth positions of poles gathered along a road segment of Vancouver, where this true data was available [43]. The calibration of the depth estimation output was performed with 50 images covering approximately 500 meters of a road. Along this segment, a total of 20 street poles were placed with known locations from the database. Different scaling factors were tested to find the optimal one minimizing the mean absolute error of the predictions.

3.2.4 Positioning

As previously stated, there can be multiple predictions of the same pole that originate from different camera positions. In order to obtain a final set of positions for the poles along a road segment, these predictions have to be combined or clustered. In this thesis, two possible solutions have been implemented. In the most straight forward way that is here referred to as ”separate predictions”, all the predictions from all the camera positions are clustered. In the second approach, ”triangulation”, pairwise matching of detections is applied in order to associate predictions for better performance.

3.2.4.1 Separate predictions

To begin with, all separate predictions are clustered together to handle multiple detections of the same object. Predictions are shown for a show case example in Figure 3.2.

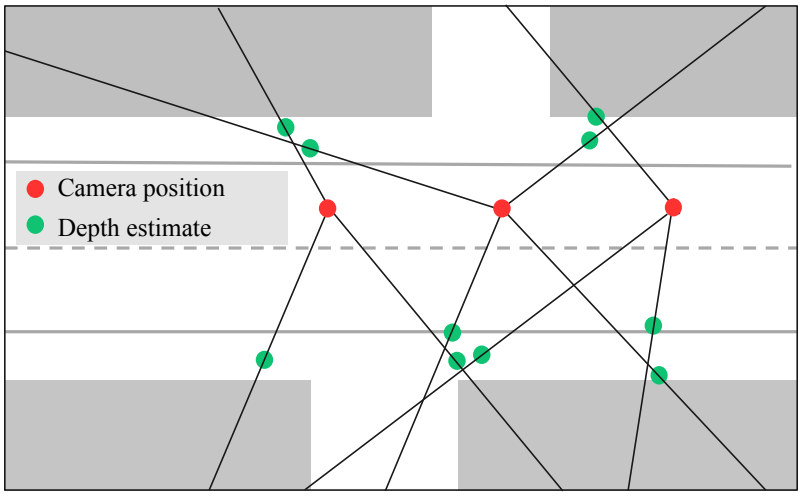


Figure 3.2: The positioning process where separate predictions in green are clustered. They are positioned from estimates of direction and distance to poles.

It is first performed one type of clustering of the separate predictions. Predictions within a specific radius are assumed to be associated and hence clustered. This is done to not include any prior knowledge of the number of predictions desired. This clustering is performed with a `sklearn` implementation called the mean shift [44]. The mean shift is an iterative method, finding the central points of the clusters that maximizes a density function [45] given all the separate predictions in total. In this implementation, a flat kernel is used with no prior knowledge of the positioning included. A clustering radius of 4 m was chosen to obtain an approximation of the appropriate number of predictions. This radius was chosen based on the true data in Kista where poles in most seen cases are separated more than 4 meters. Lowering the radius would allow poles closer each other to be detected if this would be the case, but this would also increase the risk of keeping multiple predictions of the same pole.

The mean shift clustering gives an approximation of the suitable number of clusters needed given the cluster size specified. Furthermore, it gives a set of cluster centers that are optimal for the separate predictions. However, another clustering method was found to outperform the mean shift when it comes to finding the clustering centers. Given the number of predictions from the mean shift, the separate predictions were then instead clustered through the k-means method [46]. This method iteratively chooses the predictions that from a global perspective give the least variance to the other predictions within the same cluster. The variance is here defined as the sum of squared distances between all predictions in the clusters. The `sklearn` package uses Lloyd's algorithm to iteratively find the most probable configuration [44].

From this clustering of all the predictions from all camera positions, a final set of predicted pole positions are obtained through the separate prediction method including clustering.

3.2.4.2 Triangulation

To further enhance the performance, a method for association the separate predictions has been implemented. The approach used in this thesis is inspired from [9], where possible pole locations can be approximated through triangulation. This means that each position where two lines directed from different camera locations intersect is a possible pole position. This positioning process is shown in Figure 3.3.

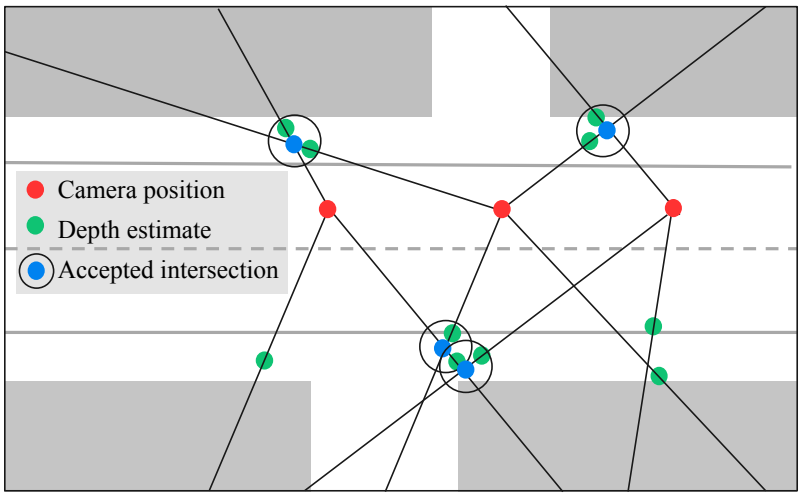


Figure 3.3: The positioning process with triangulation where accepted intersections in blue are clustered. Those have two separate predictions in green close to the associated intersection.

With multiple detections from each camera position, and multiple camera positions in proximity, the number of intersections rapidly increases. The problem then comes to choosing the most suitable intersections. The straightforward approach used for placement of trees in a similar case [9] accepts an intersection if the two depth predictions associated with the two intersecting detection lines are close enough.

This threshold has been chosen to 4.5 meters. This margin can roughly account for the given RMSE in the depth estimation. This triangulation approach is visualized with an example in Figure 3.3.

Furthermore, this positioning method can cause predictions in the vicinity of each other similar to when using the separate predictions approach. Therefore, clustering methods are applied as described as in the "separate predictions" approach section. However, this clustering is with the accepted intersections and not with all the separate predictions. It is initially a mean shift clustering, where it is assumed that there should be no poles closer than 4 meters to each other. Following this, the received number of predictions are used further in the k-means clustering method. From this, a final set of predicted pole positions are obtained through the triangulation method.

3.2.4.3 Alternative approaches

In related work, there have been approaches using only the image coordinates of the objects detected to position them. This uses the approximate camera height and an assumption of clear sight of the object and flat surface [47]. However, these assumptions do not always apply, and this approach has not provided successful results during testing.

Furthermore, other approaches have used only direction estimates. Then rules deciding how far from the road street poles are placed in general can be used for an approximate placement [48]. In a similar manner, positioning based on the placement of buildings has been implemented in [49]. Due to the difficulty of handling poles not following these rules and handling multiple detections from different camera positions, this method was not tested for this work. Similar rules applied to the height of the objects to label them in different classes could possibly also be implemented. This was only tested to a limited extent in this thesis work and could be further investigated in similar work.

In addition, methods could complement or replace the depth estimations when accepting intersections in the triangulation method. For instance, scale invariant feature matching could be applied to match features of objects [50]. However, it is difficult to implement in this case since the poles look similar to each other. Moreover, the background changes depending on the viewing direction on the pole. As another compliment to depth estimation on images, sensor input from radar could be included as well [51].

The problem of associating the predictions from different camera positions with each other has also been handled with alternative methods in related work. In [52] a probabilistic approach for data association is applied on position predictions from radar sensors. The processing of these predictions could be similar to predictions from object detection. From the predictions an estimated number of final predictions is decided, to then associate initial predictions to a final one. Due to the difficulty of associating the predictions to a specific estimate with good accuracy, and due to the complexity of the model, this was not applied in this thesis.

Finally, there have also been approaches using Markov random fields to cluster predictions as an alternative probabilistic approach [10, 53]. The concept of the random field is to iteratively test configurations from the multiple predictions to minimize a global energy function where for instance single predictions are penalized. The set of possible positions is then discrete rather than continuous as for the clustering algorithms applied in this thesis, which was regarded as limiting for the purposes of this thesis. A few test cases were performed with a random field similar to [10] without any clear improvement of the positioning.

3.2.5 Error metric

With the complete method described for positioning the poles through detecting approximate horizontal direction and distance, it is evaluated how precise the performance is. For quantification of the results, true pole positions for the area have been gathered through visual inspection of aerial and street view imagery. Each predicted pole position (x_p, y_p) can then be compared to the closest truth pole position (x_t, y_t). This can give a total mean absolute error (MAE) calculated as

$$\text{MAE} = \frac{1}{N_p} \sum_{p=1}^{N_p} \sqrt{(x_p - x_t)^2 + (y_p - y_t)^2} \tag{3.1}$$

where N_p is the total number of predictions. The mean absolute positioning error is used as an error metric in similar work [9, 10]. In addition, the ratio of matched predictions is presented as N_m/N_p where N_m is the number of predictions matched with a true position. As in [9], an upper threshold for a prediction to be regarded as a match is used, here set to 8 meters. There might be cases where a couple of predictions have the same closest true pole, but these cases are assumed to not be notably affecting the results.

3.2.6 Sources of error

There are a number of parameters that may affect the performance of the positioning algorithm. To begin with, there is an estimated general GPS accuracy of 1-5 m in 95 % of the cases [54]. This will affect the camera locations that the Google API provides, causing possible differences in the actual camera location and the given camera position. The use of further equipment in the Google cars in the form of an inertial measurement unit (IMU) should improve the accuracy to 2.5 meters according to Google [42]. The IMU combines the output of accelerometers and gyroscopes to improve the positioning of the car. The camera locations along a road follow the road in relatively straight lines that may be off from the exact line where the car is driving and also may deviate in the direction of the road. However, at least the sideways errors appear to be systematic errors causing a translation of the pole predictions. Further post processing could be applied to either manually or automatically correct the camera positioning to better handle these errors, but this has not been pursued to a greater extent within the scope of this thesis.

The positioning error of the supposed true pole positions may also affect the result. The visual inspection with a combination of aerial images with shadows and low

resolution top views of the poles and also estimation of the position from the road position in the street view images may cause an error. Especially since the timing of collecting the aerial and street view images is in some cases deviating. Visual inspection of the positions of the street light poles in the database for Vancouver [43] used for tuning the depth estimation algorithm show that they might also deviate. In Kista the assumed true positions are from visual inspection due to limited data sources. These true positions may deviate with a mean of 0.98 m which is based on comparisons with assumed true positions for about 41 of the true pole positions that are also fetched as true lighting sources in Stockholm from a database [55].

Furthermore, there are possible error contributions from the positioning algorithm itself. There may be a small deviation in the prediction of directions in the images, both seen to the assumption that the direction of the car is not exactly pointing to the next camera position and to that it should be in the exact middle of the image. However, from inspection of the images the North prediction seems reliable. Furthermore, a few of the pole detections may be with small horizontal deviations due to the limiting resolution of the panoramas. In total, these angle errors are assumed to cause negligible errors in the positioning compared to the errors caused by the GPS errors. This is since the shift of the camera position is then causing a greater deviation in the position prediction.

3.2.7 Synthetic environment

In order to isolate the possible errors caused by the positioning algorithm, the algorithm is not only applied to the collection of GSV panoramas but also to a collection of panoramas depicting a synthetic environment. The poles have been placed at what is assumed to be the true positions in the 3D environment created and visualized in Nvidia Omniverse Create as described in section 3.1. From this scenario, panoramic images picturing the synthetic world can be extracted from the exact camera positions along the road with a fish-eye lens. This results in the elimination of camera position errors and true pole position errors. For these Omniverse panoramas, the camera characteristics cause the depth estimates to be scaled with 0.35 instead of 0.25 as for the GSV panoramas. This was concluded with a similar method as for the GSV panoramas. From a few test cases in the Kista synthetic environment, the scaling factor giving the least mean absolute error to some of the true positions was chosen.

The positioning from the panoramas of the synthetic environment is varied to see how the amount of details in the environment affects the accuracy. With a fully populated model, it includes buildings with textured and colored facades as well as trees and cars. Sequentially removing the trees, then the textured facades and finally also the buildings gives a picture of the positioning algorithm's performance and sensitivity to different error contributions.

4

Results

In this chapter, the step by step results from the implementation of the workflow described in chapter 3 are presented and discussed. To begin with, the output from the image extraction is covered. Afterwards, these images are used in the object detection algorithm. Then, the obtained positions for the poles are presented. The quantified positioning result from both the real world and synthetic data are included to compare possible error contributions.

4.1 Image extraction

As described in subsection 3.2.1 panoramic images were extracted from GSV along the road Torshamnsgatan in Kista, Stockholm. For each of the 50 extracted panoramas, the camera position is provided in latitude and longitude coordinates. The camera positions are shown in Figure 4.1 overlaid on a Google Earth satellite image. Although the GSV car is driving in one of the lanes along the road when capturing the imagery, some camera positions are placed on the pavement or in the opposite driving lane. Furthermore, there can be deviations in the positions along the direction of the road. This is despite that Google performs filtering to smooth the camera positions along the road. As seen in Figure 4.1, the errors are systematic rather than randomized. Consecutive camera positions can, for instance, be shifted in the same direction. This was seen clearly when consecutive panoramas show how the real camera positions are in one of the road lanes, but where the extracted camera positions are in the middle of the road instead.

One of the extracted panoramas is shown in Figure 4.2. The corners or smaller areas of the panoramas can be distorted and some areas may be blurry because of either stitching problems or integrity reasons. The stitching problem occurs despite using stitching algorithms from [41] to stitch the extracted image tiles to a single panorama. Furthermore, sunlight, shadows and occlusion from objects in the street like cars might affect the quality of how well the panoramas cover the surrounding. Despite these issues, the area of interest in the panoramas is in most cases clear enough as seen in Figure 4.2.

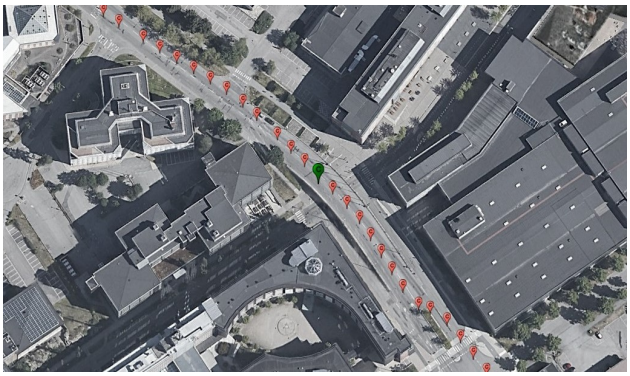


Figure 4.1: Extracted GSV camera positions shown in red overlaid on a satellite image of the area, the green marker showing the position of the panorama in Figure 4.2.

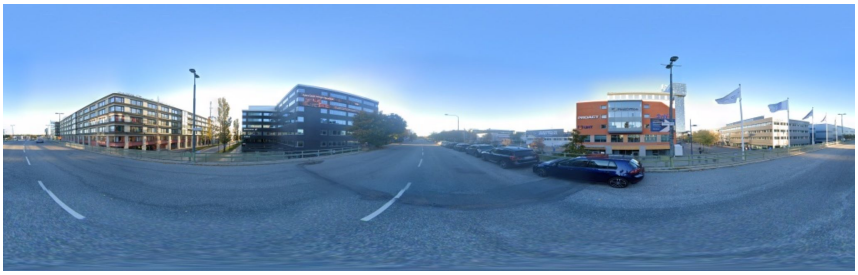
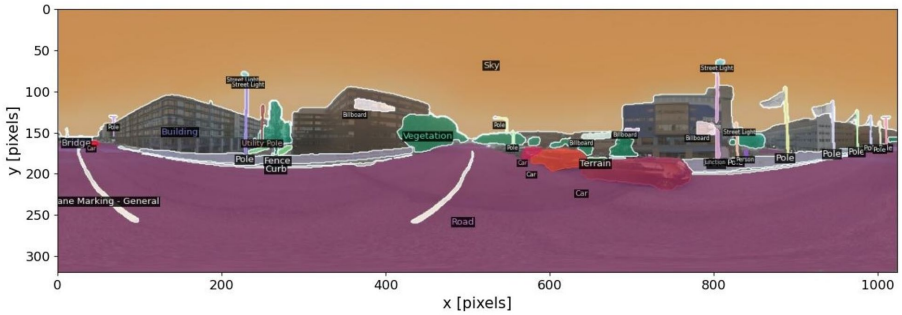


Figure 4.2: An example panorama image extracted in Kista at the position marked with a green marker in Figure 4.1.

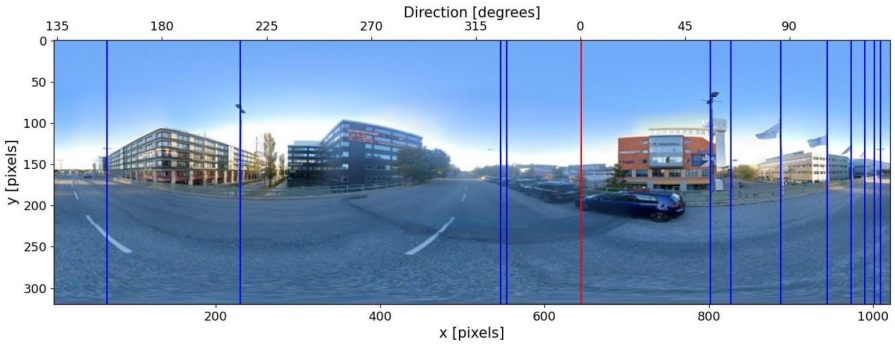
4.2 Pole detection

For each panoramic image extracted along the road in Kista the detection algorithm is applied as described in chapter 3. In Figure 4.3a the detection result from the sample panorama in Figure 4.2 is shown. The segmentation masks have been applied as a layer on top of the original image. For each separable object, the different colors show the pixels in the image associated with it and the corresponding label describes what it represents. For this thesis work, the focus is on the poles which can be seen in different colors along the road. There were a few cases when two poles were presented with the same classification number and color as if they were one pole. In these rare cases the masks were separated in post processing. This separation was performed if one pole mask contained two or more submasks with a horizontal spacing of at least one pixel between. As described in subsection 3.2.2, a pole mask is accepted if it contains at least 30 pixels to reduce the risk of multiple detections of the same pole.

From the panoptic segmentation and the extraction of pole masks, a horizontal angle is estimated for each pole mask. As seen in Figure 4.3b the estimated pole directions in blue lines match well with the detected and masked poles in Figure 4.3a. The pole directions are referenced relative to the North indicated by the red line.



(a) Panoptic segmentation.



(b) Direction estimate.

Figure 4.3: Panoptic segmentation output from the panorama image in Figure 4.2 shown as segmentation masks in Figure 4.3a and blue pole direction estimates in Figure 4.3b where red is North.

As described in subsection 2.2.4 the detected pole masks are further associated with a distance estimate in order to position the poles. The result of the monocular depth estimation is shown in Figure 4.4. The colors of the pixels represent a depth estimate described by the colorbar. For each detected pole mask with a known direction from the object detection, the associated depth values are extracted to give a depth estimate. This can be done since the depth map matrix outputted has the same dimensions as the segmentation output. As seen in the figure, the details not too distant from the camera are more clearly marked whereas the resolution of the prediction is decreasing with increasing distance. The poles closest to the camera have visible depth contours, even though the edges are not so clear. Hence, a pole prediction is accepted if the depth prediction is within 20 meters which is most often the region where the distance predictions are considered good enough. This limit is set to reduce the risk of detections that are not as qualitative as the ones at shorter distances.

For each camera position, the detections of poles are summarized and exported to a common data file. The data format of the file with comma separated values (CSV-file) is shown in Table 4.1. The index of each detection and also the camera index are noted, which are the same for the examples in Table 4.1 and thus omitted in the table. The horizontal direction to each pole given in degrees relative to true North and the depth estimate from the group of pixels being a part of the pole are noted.

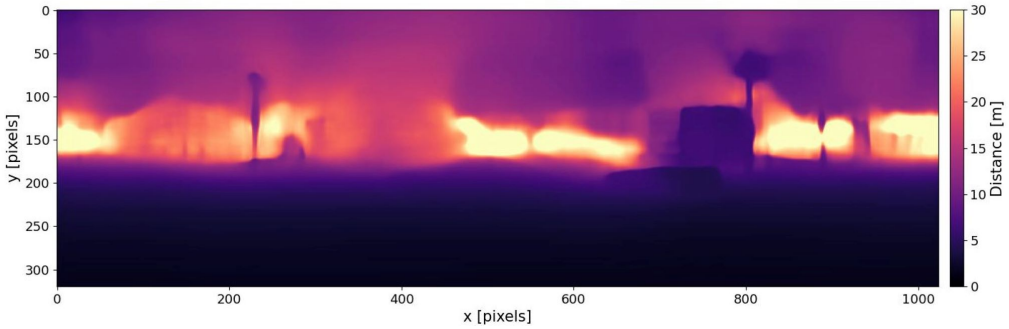


Figure 4.4: Results from the monocular depth estimation applied on the panorama image in Figure 4.2.

Table 4.1: The accepted pole detections from the generated direction output shown in Figure 4.3b and depth output shown in Figure 4.4. Direction given in angles from North, pole position in image along the x-axis in pixels and distance in meters from the camera.

Direction [degrees]	**x [pixels]**	**Distance [m]**
55.2	802	6.32
64.00	827	9.86
85.10	887	10.29
105.13	944	16.13
214.12	230	11.10

Also, the pole position in the image along the x-axis in pixels is noted in the table for comparison with Figure 4.3a. In Figure 4.3b these pole directions are shown in blue with angles given relative to the true North marked with the red line. The detections from all camera positions are gathered in the same file to be exported for input in the positioning algorithm.

Furthermore, there were a few cases where a pole was not detected by the segmentation algorithm. This was due to either the occlusion by a car, the pole being too far away such that the resolution of the image is not sufficient to present it, or because the image from the extraction was distorted in this specific area. In addition, there occurred a few false detections such that persons and tree trunks could be falsely classified as poles. Examples of those are seen in Figure 4.5. However, considering the total number of poles detected in the 50 panoramas along the street in Kista the missed and false detections were few. The mistakes were further reduced since detections were only accepted if the pole mask exceeded 30 pixels in total size in both horizontal and vertical directions. The missed detections due to occlusion or poles too far away are the reason for the reduced performance seen in the upcoming section. This is in some cases worsened considering the application of the triangulation positioning as described where at least two detections of one pole are required.

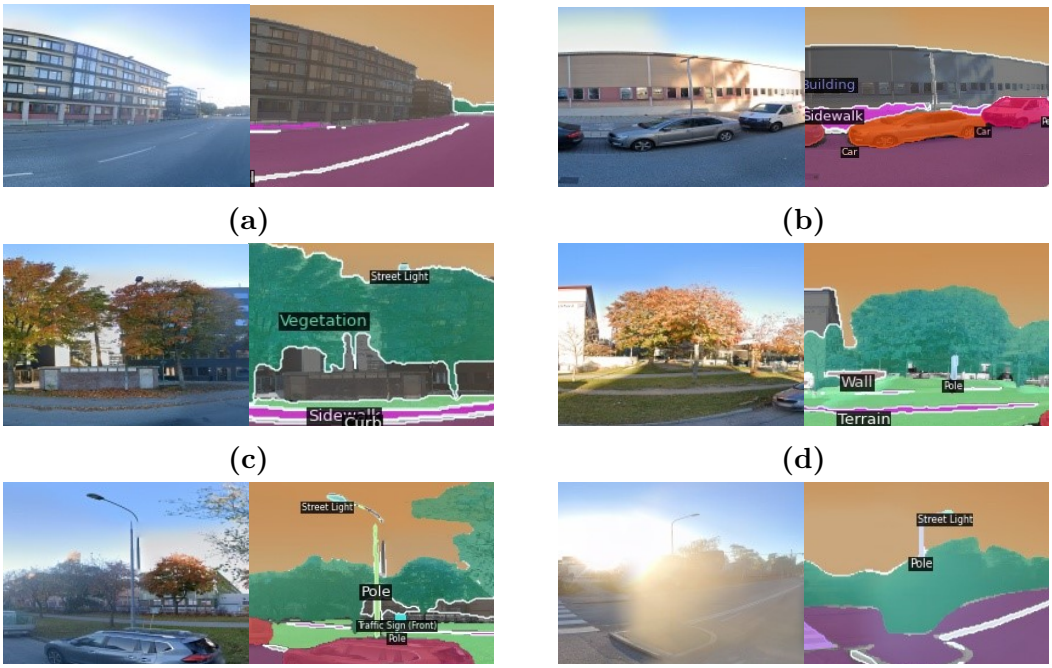


Figure 4.5: Examples of object detection difficulties. These include poles confused with the facade in Figure 4.5a and Figure 4.5b, occluded by cars or trees in Figure 4.5c and affected by distortions from the image extraction in Figure 4.5e and Figure 4.5f. Also, Figure 4.5d shows how a tree trunk is detected as a pole.

Similar to the object detection algorithm, the depth estimation can also cause a few detections with especially limiting accuracy. These were cases where it was difficult to separate the pole from the facade behind or where the object in question was too far away. In addition, sunlight and image distortions could affect the estimation.

4.3 Positioning

The detected pole direction and distance estimates are used in the two positioning algorithms described in subsection 3.2.4. First the "separate predictions" approach is applied. Secondly, the "triangulation" approach is applied. In the triangulation approach, the detections from pairwise camera positions are used to find possible intersections between rays, that are accepted if the two corresponding depth estimates are within 4.5 meters. In addition, the predictions are clustered if multiple predictions end up within 4 meters of each other. They are clustered to account for multiple detections of the same pole. The poles are assumed not to be too close to each other.

The predicted pole positions are compared to the assumed true pole positions. The true positions are obtained from visual inspection of aerial and street view images from the area. In total, there are 76 true pole positions including lamp posts, flag poles and street sign poles in the road segment. These are shown together with the

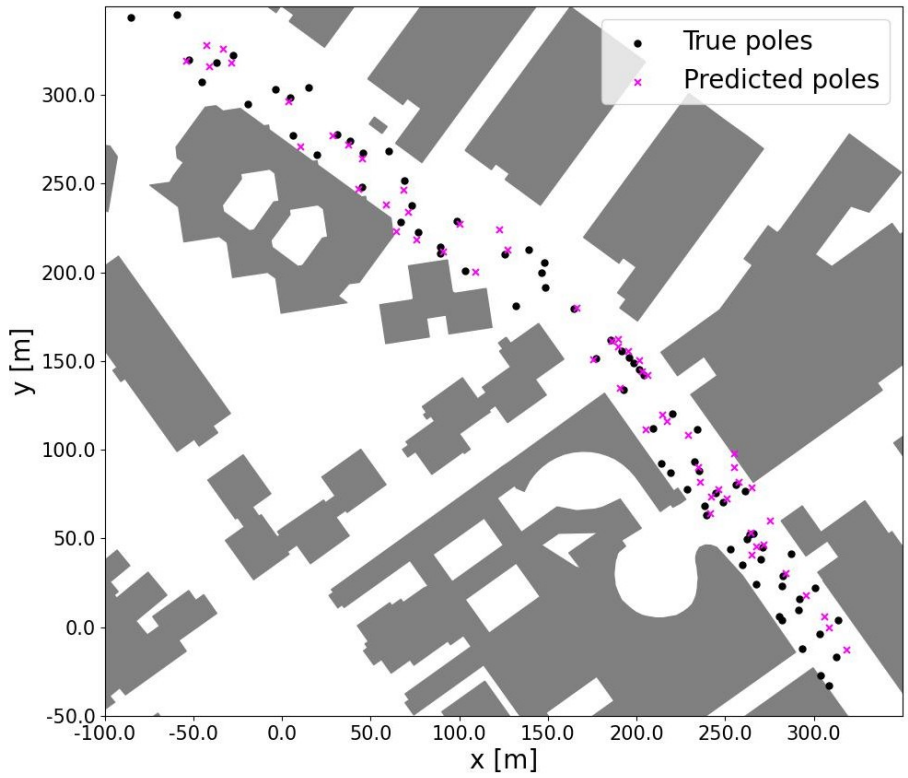


Figure 4.6: Predicted pole positions through triangulation from GSV panoramas shown as magenta crosses compared to the true positions in black along the street in Kista.

predicted pole positions from the triangulation algorithm in Figure 4.6. The CRS has EPSG code 32633 and the midpoint used as reference is chosen as $667557, 6588849$ to match the existing description of the environment.

For each predicted pole position, the distance to the closest true position is used to calculate the mean absolute error (MAE) of the predicted positions. For the detected 55 poles in the area from the triangulation method, 87% of them were matched with true values. For those matched the predicted MAE was 3.56 meters as stated in Table 4.2. The distribution of the absolute errors for this method is shown in Figure 4.7.

The result from the triangulation algorithm with the intersections used for the prediction is compared to the more simple approach separate predictions. As seen in Table 4.2 the calculated MAE is better for the triangulation method, but of similar size for both methods. Additionally, the percentage of the total predictions matched with a true pole position is lower for the separate predictions method. With this, it is meant that there are many more predictions not matched with true values causing a worse overall prediction than for the triangulation method. This is due to that all predictions are included, no pairwise matching is used to make sure a pole is detected twice to be used as a final prediction.

As a comment on this result, it has to be said that, according to the error estimation

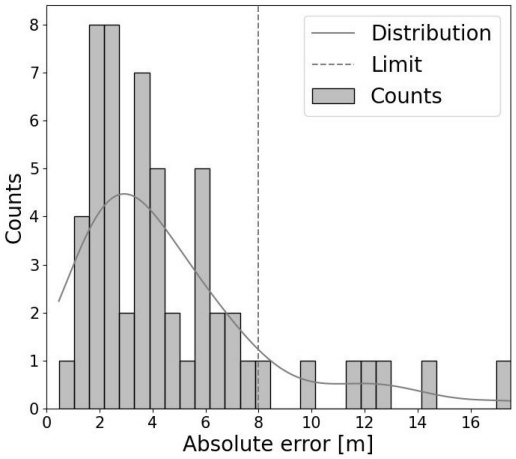


Figure 4.7: Absolute error distribution for the pole predictions from triangulation compared to the true positions along the street in Kista.

Table 4.2: Quantified performance of the positioning with single predictions and triangulation applied on GSV panoramas.

Method	Predictions	Matched pred.	MAE
Separate predictions	102	68.6 %	3.68 m
Triangulation	55	87.3 %	3.56 m

methodology described in subsection 3.2.5, the GPS accuracy of the extracted camera positions is approximately 2.5 meters. Additionally, the assumed true positions are placed from visual inspection of aerial and GSV images leading to an approximate error contribution that should be small but also could also have an effect. Furthermore, errors may occur due to missed and false detections as in Figure 4.5. Further, there might be multiple detections of the same true pole position. In addition, poles to the lower left are further away than 20 m from the lane where the car is driving, which is outside the region where the detections are good enough. This is a shortcoming of the current prediction and could maybe be improved if images were captured from both lanes of the road. Additionally, a shorter distance between the camera positions and thus more detections could possibly improve the results.

4.4 Synthetic environment

Finally, the poles are included in a 3D environment used as a base for the radio propagation simulations. This 3D environment is created from a combination of the terrain data of an area in combination with shapefiles including the outlines of buildings and street networks as described in section 3.1. In the created 3D environment the poles have been added at the true pole positions, see Figure 4.8b. While moving the camera in the 3D world, panoramic images are extracted with headings matching the existing real world panoramas.

(a) GSV panorama.

(b) Omniverse panorama.

Figure 4.8: Corresponding panorama images from GSV in Figure 4.8a and the synthetic Omniverse environment in Figure 4.8b with poles at the true positions.

In the most complex synthetic 3D environment, there are detailed buildings, a street network, a vegetation layer and of course the poles at the true positions. In this synthetic environment, panoramas are extracted exactly at the assumed camera position. This enables testing of the positioning algorithm while the error contributions from camera positions and true pole positions are eliminated, as described in subsection 3.2.7. In Figure 4.8 a sunny sky in the synthetic environment is displayed to match the GSV image. However, for the positioning a gray sky has been used to avoid the effect of distracting sunlight and sharp shadows.

The fully detailed environment includes the street network, buildings with facades with clear textures and a foliage layer with trees placed. This will be referred to as case C1. In further tests, the trees are first removed (case C2), followed by the removal of the textured facades (case C3) and finally also the buildings are removed (case C4). The extraction of panoramic imagery from the synthetic environment, and application of the positioning algorithm, is performed for all cases. The positioning results are presented in the form of MAE and percentage of matched predictions as described in Table 4.3.

As seen in Table 4.3, the positioning algorithm gives the best result when the synthetic environment does not include trees in case C2. The result from the positioning algorithm based on this data is shown in Figure 4.9. By inspection of the detection output, this is mostly due to two reasons. The first one is the trees leading to occlusion of the poles such that they are not detected. The second one is the similarity of the tree trunks to poles in this synthetic environment which confused the object detection algorithms. These missed and false detections in case C1 showed to cause more and worse positioning predictions compared to case C2.

Table 4.3: Quantified performance of the positioning algorithm applied on Omniverse panoramas showing environments with varying levels of details.

Case	Data	Predictions	Matched pred.	MAE
C1	Fully detailed	73	91.7 %	1.38 m
C2	No vegetation	67	98.5 %	0.94 m
C3	No facade texture	74	95.9 %	1.69 m
C4	No buildings	83	80.7 %	2.68 m

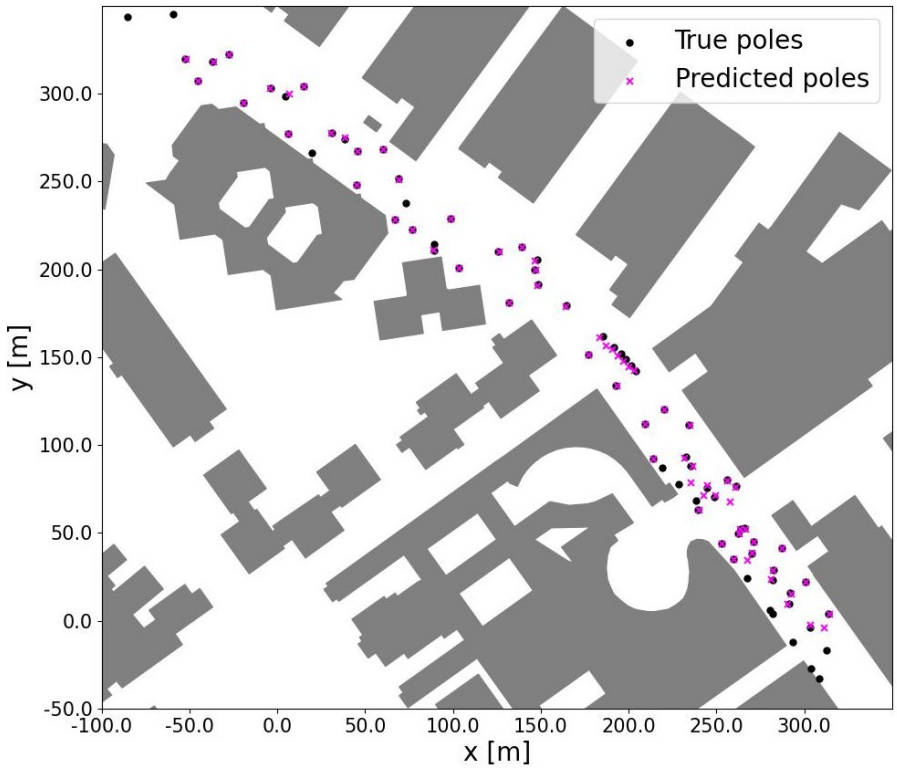


Figure 4.9: Predicted pole positions from Omniverse panoramas of an environment without trees as shown as magenta crosses compared to the true positions in black along the street in Kista.

There is a slight increase in errors when removing also the more textured and detailed facades in case C3. In the last case (C4) when the buildings were removed the errors increased even more. This is mostly due to problems with the depth estimation algorithm. Since the MDE algorithm was trained on real world scenarios, the differences compared to this relatively empty environment may have been too big, leading to bad quality predictions. This was seen with difficulties handling for instance street crossings as well as separating the specific distance to poles against the background. The empty environment in case C4 did however make it easier for the object detection algorithm to detect poles on a longer range that were not detected before.

It should be noted that the application of the positioning algorithm on the synthetic environment indicates that the positioning error of the camera positions and true positions have a relatively big influence on the result. However, the similarity to the real GSV scenarios is not perfect since there are details that are not included in the synthetic scenarios. To name a few examples, the real world includes a greater number of cars, more extensive vegetation and of course also people, bicycles and fences around buildings or bridges. In addition, the synthetic poles may be easier to detect than real poles both due to the pole design and the surrounding environment.

5

Simulation

In this chapter, an introduction is presented of the performed ray tracing simulations. This is followed by the implemented simulation scenario with the enriched 3D environment based on results in chapter 4. Afterwards, an analysis of the impact on the channel characteristics from the enrichment is presented.

5.1 Ray tracing tool

The aim of the ray tracing simulations is to investigate the impact of the detected and positioned poles on the characteristics of the radio propagation. The simulations are performed with a ray tracing tool developed by Ericsson. As described in section 3.1, a 3D environment saved as a USD file can be created for the desired area in Stockholm. This can be used as a base for the simulation environment of the ray tracing tool.

To use the environment in the simulation tool, the electromagnetic characteristics of each 3D object in the environment is modeled. This is done according to radio propagation theory that is briefly described in section 2.1. Most importantly the buildings are modeled with suitable surface characteristics to account for correct reflection, scattering and diffraction. Reflection from the ground is also included. In addition, areas with vegetation are assumed as a medium causing the rays to attenuate. The developed scattering model for poles has been implemented based on the radio propagation theory introduced in subsection 2.1.3 following [7].

5.2 Simulation scenario

A simulation scenario is prepared by specifying a position for the transmitters and the receivers. As introduced in section 3.1, the scenario in focus for this thesis is a part of Kista, Stockholm, where measurements have been performed. The measurements interesting for this work are more thoroughly described in [5].

One of the base stations in the Kista scenario is placed on the roof of a building, with approximate coordinate $(-163, 483, 56)$ from the map center. The positions are given according to a coordinate system referenced from the center of the chosen 2000 times 1800 meters area. During the measurements, the receiving antenna is

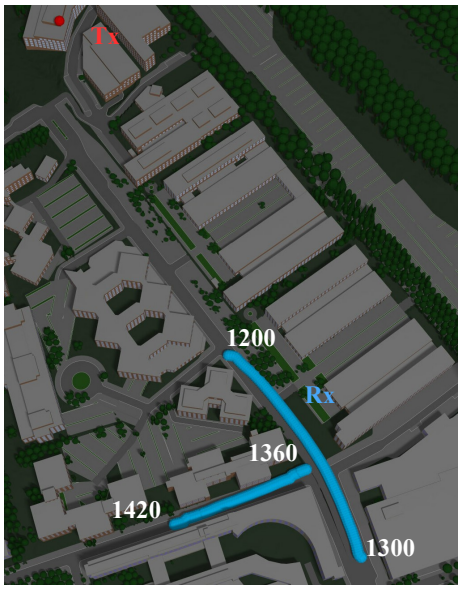


Figure 5.1: The traces of receiver indices in Kista from the selected measurements. Section S1 with trace index between 1200 and 1300 is along Torshamnsgatan and the orthogonal section S2 from 1360 to 1420 is along Kistagången.

mounted on a moving van driving along the streets of Kista, covering a big area. Two segments of the track are used in this thesis. These are the sections with receiver index 1200 to 1300, referred to as section one (S1), and with receiver index 1360 to 1420 (S2) as displayed in Figure 5.1. To match the measurements in the simulations, the receiver is moved along these positions.

These two sections S1 and S2 are focused on, as here the analysis showed the clearest indications of poles affecting the measurements have been experienced. Along S1, poles placed close to this road segment are believed to affect the measurements clearly observed by the Doppler effect. When the poles are passed with the measurement van, the scattering from the poles give a measured Doppler shift as introduced in section 2.1. This is especially clear in S1 since the poles are in LOS propagation from the transmitter and the receiver and since there is not that much vegetation in the vicinity.

In section S2 from receiver index 1360 to 1420 the interest is instead in analysing the around the corner scattering. Here, the same poles at the road segment along S1 are believed to have effect on the around the corner scattering. Then, the quite strong LOS propagation from the transmitter reaches the poles in segment S1 that then scatter some of the power around the corner into segment S2.

For the inclusion of the poles in the simulations, only the poles in road segment 1200 to 1300 in Figure 5.1 are included. This is since they are assumed to be the ones contributing the most to the effects described above including the Doppler shift in S1 and around the corner scattering to S2. Initially, the true pole positions are used. These were anyway gathered for evaluating the positioning algorithm and can give a general picture of the effect from enrichment of 3D scenarios without effect

Table 5.1: The most important parameters for the ray tracing simulation. These include the center frequency and bandwidth, sampling and interaction specifications as well as pole dimensions.

Parameter	**Value**
Center frequency	2.66 GHz
Bandwidth	20 MHz
Sampling interval	1/187.5 s
Consecutive samples	101
Max. tot. interactions	6
Max. specular reflections	4
Max. diffractions	2
Max. diffuse scatterings	1
Pole height	10 m
Pole width	0.4 m
Pole section length	0.25 m

from positioning errors. However, the detected poles from the same area are also extracted for comparison. As noted in chapter 4, most of the poles are detected and these deviate a few meters from the true positions.

For the simulation scenario the details about the settings are also specified in Table 5.1. These are matched to the measurement scenario. In addition to pole positions, the specifications include the orientation and tilt of the transmitting antenna, the gain of the antennas, and the center frequency and bandwidth. The latter are set to 2.66 GHz and 20 MHz, respectively. Furthermore, the number of interactions allowed between a transmitting and receiving point is limited, see Table 5.1. This is due to the otherwise exponentially increasing number of paths and computational time required to include contributions that could be seen as negligible. Also the pole characteristics are specified, where each pole has a height of 10 meters and width of 0.4 meter. This is slightly exaggerated compared to reality to clearly be able to visualize their impact. In addition, the poles are divided into 40 different scattering pole sections of length 0.25 meter. The most important simulation parameters are gathered in Table 5.1.

To summarize, possible improvements regarding making the digital twin more realistic are investigated by comparing measurements to simulations with and without the inclusion of the poles in this Kista scenario. This is done by looking at changes in the frequency shift due to the Doppler effect along the street in section S1 from receiver index 1200 to 1300 and path gain from eventual around the corner scattering in section S2 from receiver index 1360 to 1420.

5.3 Signal processing

The measurements collected for the area are in the frequency domain for a limited bandwidth. This means that the results are obtained as time-varying channel transfer functions $H(f,t)$ at each individual measurement location along the track. This is called the channel transfer function (CTF) and models the output as a function of frequency and time for the configuration of measured signals.

The output of the ray tracing tool includes the geometric paths of the rays traveling from the transmitter to the receiver for the different positions. The paths are calculated given LOS propagation and the interaction properties defined. Furthermore, the simulation results of the ray tracing tool provide the propagation parameters in the channel impulse response $h(\tau,t)$ expressed in Equation 2.3. For comparison with the measurements, the CTF in the frequency domain has to be calculated. The band limited time varying CTF is calculated via Fourier transform from the delay domain in Equation 5.1. Here, the number of frequency samples and bandwidth has been matched with the measurements.

$$H(f,t) = \mathcal{F}_\tau\{h(\tau,t)\} \tag{5.1}$$

From the CTF of the measurements and simulations, respectively, the path gain PG can be estimated for each individual receiver location. This is calculated by averaging over the frequency and time of the samples, noted with an expectation value as

$$PG = \mathbb{E}(|H(f,t)|^2)\ . \tag{5.2}$$

The averaging is performed in order to reduce the small scale fading effect which occurs due to coherent summation of the individual complex valued path gains in the impulse response of Equation 2.3. Small scale fading occurs when paths have similar strength but opposite phases and thus cause destructive interference [15].

Furthermore, the Doppler power profile can also be estimated from the CTF of the measurements and simulations, respectively. Prior these calculations, a Hann window function was applied to the time domain to suppress the effect of sidelobes. Then, an inverse Fourier transform is applied on the time domain to instead express the output as a Doppler variant transfer function. Finally, a summation over the discrete frequency samples is performed to obtain a final Doppler power profile as a function of only the Doppler shift ν.

$$D(\nu) = \sum_f \mathcal{F}_t^{-1}\{H(f,t)\} \tag{5.3}$$

5.4 Impact on channel characteristics

Finally, the results for the ray tracing simulations are presented. The ray tracing tool output enables both investigation of the channel characteristics and a visualization of the rays traveling from the transmitter to the receiver. Before going into the

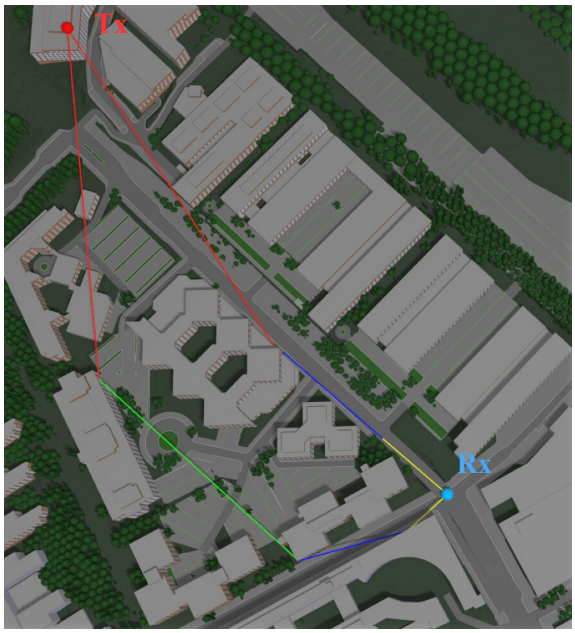


Figure 5.2: A couple of rays with multiple interactions traced between transmitter in red on top of a building and receiver in blue placed at the streets of Kista, Stockholm.

changes in the channel characteristics, an example of rays propagating in one of the Kista simulations is shown in Figure 5.2. There, one transmitter and one receiver have been specified to the simulating tool that then outputs the traces of the rays between them, colored after how many interactions such as diffraction or reflection they have experienced.

5.4.1 Doppler shift

As described in section 5.1 the scattering model for poles have been implemented on top of the previous simulation environment. Based on the superposition principle, it is thus possible to compare simulation results including poles to simulations without poles. The poles included in the ray tracing simulations are here the poles limited to the chosen section of receiver positions between index 1200 and 1300. The poles in this section are of most interest for analyzing the Doppler shift.

For the chosen receiver positions in section 1200 to 1300 the power spectrum is presented in Figure 5.3. As the receiver index is changing, there is a Doppler shift of the frequency. This is due to the velocity of the receiver relative to the static objects such as buildings, the poles and the transmitter. In Figure 5.3, the normalized power spectrum is shown for the measurements and for the simulations without and with poles, respectively. In addition, the contribution from only the poles based on the simulations is shown for clarity. The power spectrum has been normalized for each case individually.

The brighter line with negative Doppler frequency in the power spectrum with the highest relative power is the LOS path directly from the transmitter. This has a

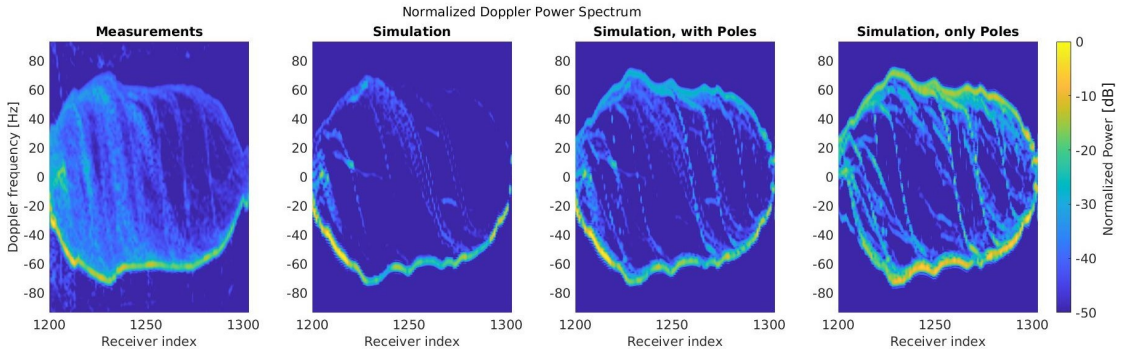


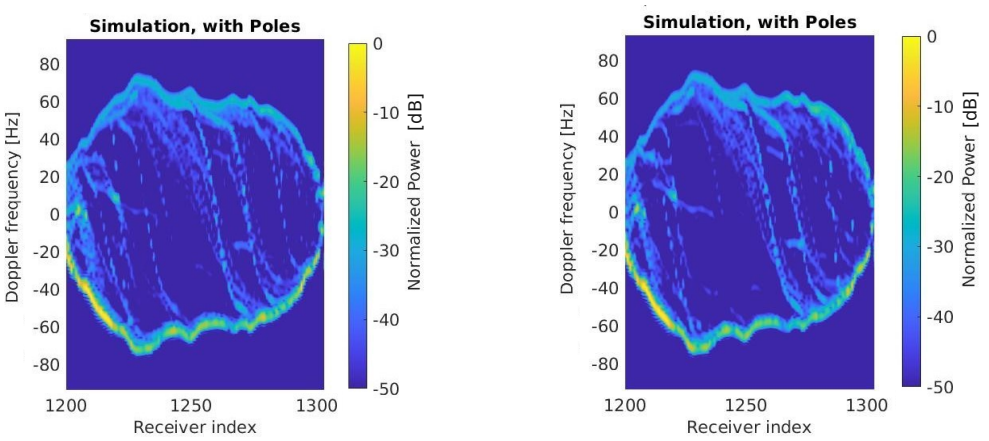
Figure 5.3: Normalized power spectrum showing the Doppler frequency for receiver index 1200 to 1300. From left to right, the plots show results for measurements, simulations without poles, simulations with poles, and simulations with only poles.

negative Doppler frequency since the receiver is moving away from the transmitter. The Doppler frequency is changing with the velocity of the receiver since the Doppler shift is higher at a greater speed. In addition, the relative angle between the receiving wave direction and the direction of movement influences the Doppler shift as introduced in Equation 2.4. The indications of positive Doppler frequency is due to backscattering. These are waves propagating to and reflecting on objects that the receiver is traveling towards. As the receiver is passing an object, an inverse "S"-shaped Doppler shift of the power associating to this object is seen.

In the measurement spectrum in the left-most plot in Figure 5.3, there are indications of increased power at a few places with higher absolute Doppler frequency than what would be obtained due to the speed of the car. This may be due to other cars driving in the opposite direction of the van on which the receiver is mounted. As a result, the relative velocity and hence the Doppler frequency is increased.

In the simulation spectrum with poles included, it can be seen that there is more backscattering than in the simulation without poles. Furthermore, there are more of the inverse "S"-shaped shifts from positive Doppler frequency to negative. This indicates that the receiver is passing the poles and thus changing rapidly from positive to negative Doppler. This can clearly be seen in the spectrum. This can also be seen in the right most plot in Figure 5.3 which shows the contribution only from the poles. The reason for the bright color of this plot is as mentioned that the color scales are normalized individually for each plot. The normalized spectrum shows that the pole contributes with a relative evenly spaced power spectrum over positive and negative Doppler frequency. This may be due to that the poles are quite evenly spaced out along the trace traveled by the receiver, resulting in that there are poles both in front of and behind the moving receiver.

Furthermore, the distances between the inverse "S"-shapes are quite similar seen to the measurement and simulation results. This indicates that the poles have relative to each other the right spacing. In addition, they pass through zero at approximately similar receiver index. Thus, the positioning of the poles is perceived as quite similar.



(a) True pole positions. **(b)** Detected pole positions.

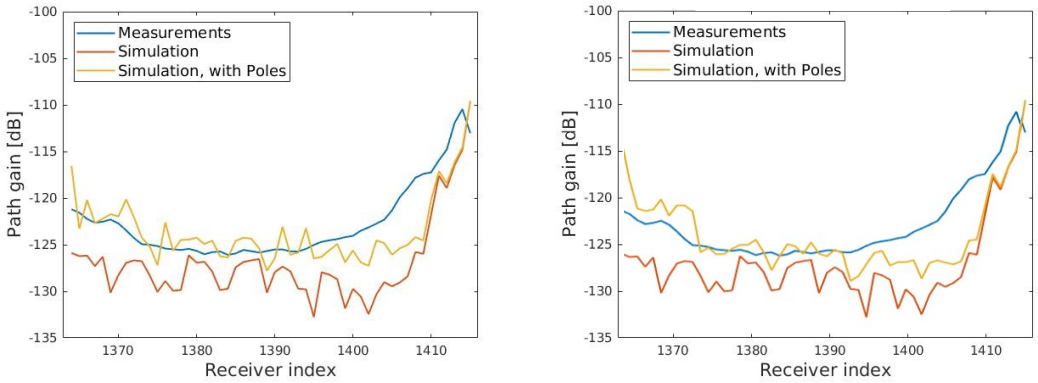
Figure 5.4: Normalized power spectrum showing the Doppler frequency for receiver index 1200 to 1300. In Figure 5.4a the true pole positions in the section have been used, in Figure 5.4b the detected ones.

As a further comparison, the results from similar simulation performed with the detected poles in the same section is showed in Figure 5.4b. There is some differences between the normalized Doppler power spectra, but in general these two are quite similar. This would mean that the general effect on the channel characteristics is similar, even though the detected poles do not exactly match the number of true poles nor their the exact placement.

5.4.2 Path gain

For the around corner investigation, the interesting road section is S2 with receiver index 1360 to 1420 as shown in Figure 5.1. The path gain for this section is presented in Figure 5.5. This is the path gain calculated for the measurements and simulations without and with including the poles, respectively. The path gain has been calculated through coherent combination of the phase from the contributing rays. In Figure 5.5a the true pole positions have been used and in Figure 5.5b the detected ones. As can be seen, there is an increased path gain level when the poles are included. The increase is due to around the corner scattering from the poles. The results are also getting closer to the measurements when including poles. There are some differences between the effects from the true and detected pole positions, but as in the case with the power spectrum the characteristics are in general similar.

It should be noted that there are specified simulation parameters that are assumed to match the measurement scenario, but where small deviations still could affect the results. It can be difficult to match the antenna properties such as tilt and position exactly, which is affecting both the Doppler power spectrum in S1 and the path gain in S2. There could also be small deviations in the positions and heights of buildings having effect on LOS propagation paths. Furthermore, vegetation may obstruct the LOS path in the measurements whereas in simulations a simplified vegetation may not be sufficient. In general, a comparison between the simulation and measurement



(a) True pole positions. **(b)** Detected pole positions.

Figure 5.5: Path gain for the section of receiver indices in the orthogonal street, showing an increase due to scattering from the poles around the corner. True pole positions in Figure 5.5a and detected positions in Figure 5.5b.

indicate fewer differences when including the poles.

It can here be seen that in general the inclusion of the poles has led to simulation results more similar to the measurement results. In other words, the enrichment of the simulation environment has made the channel characteristics more realistic.

6

Conclusion

In this last chapter, the conclusions from this thesis work are presented. This will include a summary of the contribution from the work, possible improvements of the workflow and also an outlook covering future opportunities.

6.1 Summary

In this thesis, a positioning process including extraction of street view images, object detection, monocular depth estimation and clustering has been applied to enrich a 3D environment with street poles. The street view images were extracted from Google Street View (GSV) in panorama format. In each panorama, the North was approximated. The poles were detected with panoptic segmentation to obtain a horizontal direction for each pole. In addition, the pole masks were matched to the Monocular Depth Estimation (MDE) output for a distance estimate. Combining distance and direction detections from multiple camera positions was used to cover a whole road segment. All separate predictions were used in an initial approach, but an additional method with triangulation increased the performance. The predictions were clustered to avoid multiple predictions of the same pole. First with mean shift clustering given a cluster radius to result in a suitable number of clusters. This was followed by k-means to get cluster centers given the number of clusters.

The final positioning was achieved with a mean absolute error of 3.5 meters with an 87 % match to true values for a specific test case in Stockholm. In addition, the number of poles detected for this test case is close to the true value. The test case included 76 true poles of street signs and street lights along a road section where 50 panorama images were extracted with approximately 10 meters apart. The prediction includes a few double detections and a few missing detections due to limitations of the detection at longer distances.

Investigations were also performed with a synthetic environment to investigate contributions from different error sources. The investigation indicated that the major contribution to the errors in the positioning comes from limitations in accuracy for camera positions and uncertainties concerning true pole positions. Eliminating these two contributions in the synthetic scenario provided an improved accuracy of the positioning algorithm to an MAE of 0.9 meters with a 98 % match to true values.

Furthermore, this work indicates that inclusion of scattering objects for street furniture has a non-negligible effect on the channel characteristics of a digital twin for a radio network. More specifically, the effect on the Doppler frequency and channel richness observed in Doppler power profiles is improved when compared to measured results. In addition, a contribution to the path gain is observed from pole scattering around the corners of buildings. The enriched 3D environment can be regarded as an important contribution to making the digital twin of the radio network more realistic.

6.2 Improvements

There are some improvements that should be mentioned for the enrichment of the environment. The clearest improvement includes correcting the positioning of the camera. Investigations of the synthetic scenarios eliminating these parameters give much better results. Two possible ways can be envisioned. One is considering the camera positions as random variables and optimizing them jointly with the pole prediction. Secondly, the camera positions could be improved by using the information in the extracted image to correct the given position. This could be developed in a close to automatic manner by exploiting for instance building location and height or street position in an image. The locations of the building and streets are known to some degree of accuracy, and positioning of poles relative to those could possibly be performed.

In addition, the use of the synthetic environment could possibly be further extended. This could include for instance including more details but also investigating further the effect of parameters such as the distance between camera positions, the weather or the appearance of the included poles. It is also possible to make use of other information such as the alignment of the road in the images for comparisons with the real world images.

Furthermore, it could be investigated how the object detection and depth estimation algorithms would perform with variations of the input parameters. One obvious parameter would be the image resolution and also image type, regarding that for instance images with a more narrow field of view than panoramas could be used as well. It could also be possible to train new models more suitable for detecting the direction and depth estimations of poles specifically. Applying transfer learning with created datasets to further train the current models would probably be an interesting approach.

Additionally, it could be interesting to make use of a more extensive data collection if similar measurements are performed in the future. Collecting image material for the track of the receiver position is one possibility that could improve the camera positioning problem. Then the capture time for radio measurement and images would match better as well. Further, collecting lidar data for range estimation could be an option to use as either a compliment for the monocular depth estimation or for an improvement of the calibration of it.

On a more technical level, the pipeline from input parameters to an outputted result could also be automated to a greater extent than the current status. The number and format of input parameters could be improved to be easier to specify. Also, the data extraction output could go immediately to the object detection and depth estimation algorithms and finally to the positioning algorithm for inclusion in the 3D environment without intermediate manual steps such as running different scripts.

6.3 Outlook

Finally, there could be possibilities that the work presented in this thesis could be useful for future work with a similar topic. One possibility would be to extend the model to include areas of an urban environment instead of only a road. This would simplify the input parameter format, as well as give a better coverage and sample of the city as a whole rather than only analyzing one chosen road. In this way, a more extensive model to approximate the distribution and type of poles in a specific type of city could be developed and utilized for a more extensive 3D model generation.

Moreover, digital twins for ray tracing simulations are in constant development, making it possible to enrich the 3D environment even further in the future. This could include scattering models for other types of street furniture such as trees, parked cars and benches to name a few examples. To design a similar positioning method for extracting, detecting and positioning these objects would be an interesting continuation of the work in this field.

Bibliography

[1] D. He, B. Ai, K. Guan, L. Wang, Z. Zhong, and T. Kürner, "The design and applications of high-performance ray-tracing simulation platform for 5G and beyond wireless communications: A tutorial," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 10–27, 2018.

[2] Z. Yun and M. F. Iskander, "Ray tracing for radio propagation modeling: Principles and applications," *IEEE Access*, vol. 3, pp. 1089–1100, 2015.

[3] D. Jones, C. Snider, A. Nassehi, J. Yon, and B. Hicks, "Characterising the digital twin: A systematic literature review," *CIRP Journal of Manufacturing Science and Technology*, vol. 29, pp. 36–52, 2020.

[4] Wikipedia, "Street furniture." `https://en.wikipedia.org/wiki/Street_furniture`, 2022. [Online; accessed 2023-06-05].

[5] H. Asplund, M. Johansson, M. Lundevall, and N. Jaldén, "A set of propagation models for site-specific predictions," in *12th European Conference on Antennas and Propagation (EuCAP 2018)*, pp. 1–5, 2018.

[6] B. Göktepe, M. Peter, R. J. Weiler, and W. Keusgen, "The influence of street furniture and tree trunks in urban scenarios on ray tracing simulations in the millimeter wave band," in *2015 European Microwave Conference (EuMC)*, pp. 195–198, IEEE, 2015.

[7] Guan, Ke et al., "On the influence of scattering from traffic signs in vehicle-to-x communications," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 8, pp. 5835–5849, 2015.

[8] D. Chizhik, J. Du, and R. A. Valenzuela, "Comparing power scattered by RIS with natural scatter around urban corners," in *2022 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting (AP-S/URSI)*, pp. 1606–1607, IEEE, 2022.

[9] S. Lumnitz, T. Devisscher, J. R. Mayaud, V. Radic, N. C. Coops, and V. C. Griess, "Mapping trees along urban street networks with deep learning and street-level imagery," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 175, pp. 144–157, 2021.

[10] V. A. Krylov, E. Kenny, and R. Dahyot, "Automatic discovery and geotagging of objects from street view imagery," *Remote Sensing*, vol. 10, no. 5, p. 661, 2018.

[11] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1290–1299, 2022.

[12] G. Neuhold, T. Ollmann, S. Rota Bulo, and P. Kontschieder, "The Mapillary Vistas dataset for semantic understanding of street scenes," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4990–4999, 2017.

[13] A. F. Molisch, *Wireless communications*. John Wiley & Sons, 2012.

[14] N. Al-Falahy and O. Y. Alani, "Millimetre wave frequency band as a candidate spectrum for 5G network architecture: A survey," *Physical Communication*, vol. 32, pp. 120–144, 2019.

[15] A. B. Smolders, H. J. Visser, and U. Johannsen, "Modern antennas and microwave circuits - A complete master-level course." `https://arxiv.org/abs/1911.08484`, 2022. [Online; accessed 2023-03-31].

[16] D. Astely, P. Von Butovitsch, S. Faxér, and E. Larsson, "Meeting 5G network requirements with Massive MIMO," *Ericsson Technology Review*, vol. 2022, no. 1, pp. 2–11, 2022.

[17] M. I. Skolnik, "36 - radar," in *Reference Data for Engineers (Ninth Edition)* (W. M. Middleton and M. E. Van Valkenburg, eds.), pp. 36–1–36–22, Woburn: Newnes, ninth edition ed., 2002.

[18] P. Series, "Effects of building materials and structures on radiowave propagation above about 100 mhz," *Recommendation ITU-R*, pp. 2040–1, 2015.

[19] V. Degli-Esposti, "A diffuse scattering model for urban propagation prediction," *IEEE Transactions on Antennas and Propagation*, vol. 49, no. 7, pp. 1111–1113, 2001.

[20] D. A. McNamara, C. W. Pistorius, and J. Malherbe, *Introduction to the uniform geometrical theory of diffraction*. Artech House Norwood, MA, 1990.

[21] V. DiCaudo and W. Martin, "Approximate solution to bistatic radar cross section of finite length, infinitely conducting cylinder," *IEEE Transactions on Antennas and Propagation*, vol. 14, no. 5, pp. 668–669, 1966.

[22] Y. L. de Jong and M. H. Herben, "A tree-scattering model for improved propagation prediction in urban microcells," *IEEE Transactions on Vehicular Technology*, vol. 53, no. 2, pp. 503–513, 2004.

[23] K. O'Shea and R. Nash, "An introduction to convolutional neural networks." `https://arxiv.org/abs/1511.08458`, 2015. [Online; accessed 2023-03-10].

[24] L. Deng, "The MNIST database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[25] S. An, M. Lee, S. Park, H. Yang, and J. So, "An ensemble of simple convolutional neural network models for MNIST digit recognition." `https://arxiv.org/abs/1511.08458`, 2020. [Online; accessed 2023-03-10].

[26] Lin, Tsung-Yi et al., "Microsoft COCO: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755, Springer, 2014.

[27] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[29] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[30] W. Abdulla, "Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow." `https://github.com/matterport/Mask_RCNN`, 2017. [Online; accessed 2023-02-10].

[31] Liu, Ze et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021.

[32] O. Elharrouss, S. Al-Maadeed, N. Subramanian, N. Ottakath, N. Almaadeed, and Y. Himeur, "Panoptic segmentation: A review." `https://arxiv.org/abs/2111.10250`, 2021. [Online; accessed 2023-03-20].

[33] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2." `https://github.com/facebookresearch/detectron2`, 2019. [Online; accessed 2023-02-20].

[34] M. Meyer and G. Kuschk, "Automotive radar dataset for deep learning based 3D object detection," in *2019 16th European Radar Conference (EuRAD)*, pp. 129–132, IEEE, 2019.

[35] A. Masoumian, H. A. Rashwan, J. Cristiano, M. S. Asif, and D. Puig, "Monocular depth estimation using deep learning: A review," *Sensors*, vol. 22, no. 14, p. 5353, 2022.

[36] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3828–3838, 2019.

[37] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[38] Anguelov, Dragomir et al., "Google street view: Capturing the world at street level," *Computer*, vol. 43, no. 6, pp. 32–38, 2010.

[39] Google Earth Pro ver. 7.3, "Torshamnsgatan Kista 59°24'22.00"N 17°57'17.91"E, elevation 225 m." [Online; accessed 2023-04-27].

[40] G. Spaias, "Facade Feature Extraction." `https://github.com/Georspai/Facade_Feature_Extraction`, 2023. [Online; accessed 2023-02-14].

[41] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[42] Google, "Street view ready pro specifications." `http://aiweb.techfak.uni-bielefeld.de/content/bworld-robot-control-software/`, 2019. [Online; accessed 2023-04-13].

[43] City of Vancouver, "Open Data Portal, Street lighting poles." `https://opendata.vancouver.ca/explore/dataset/street-lighting-poles/information/`, 2023. [Online; accessed 2023-03-22].

[44] Pedregosa, F. et al., "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[45] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[46] S. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[47] S. Branson, J. D. Wegner, D. Hall, N. Lang, K. Schindler, and P. Perona, "From Google Maps to a fine-grained catalog of street trees," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 135, pp. 13–30, 2018.

[48] C. Zhang, H. Fan, and W. Li, "Automated detecting and placing road objects from street-level images," *Computational Urban Science*, vol. 1, pp. 1–18, 2021.

[49] D. Laumer, N. Lang, N. van Doorn, O. Mac Aodha, P. Perona, and J. D. Wegner, "Geocoding of trees from street addresses and street-level images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 162, pp. 125–136, 2020.

[50] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[51] V. A. Krylov and R. Dahyot, "Object geolocation using MRF based multi-sensor fusion," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 2745–2749, IEEE, 2018.

[52] M. Lundgren, L. Svensson, and L. Hammarstrand, "Variational bayesian expectation maximization for radar map estimation," *IEEE Transactions on Signal Processing*, vol. 64, no. 6, pp. 1391–1404, 2015.

[53] J. D. Wegner, S. Branson, D. Hall, K. Schindler, and P. Perona, "Cataloging public objects using aerial and street-level images-urban trees," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6014–6023, 2016.

[54] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *Computer*, vol. 34, no. 8, pp. 57–66, 2001.

[55] Stockholms Stad, "Belysningsmontage, Öppna data." `https://dataportalen.stockholm.se/dataportalen/GetMetaDataById?id=LvFeature13185471&showmetadataview`, 2023. [Online; accessed 2023-05-05].

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY