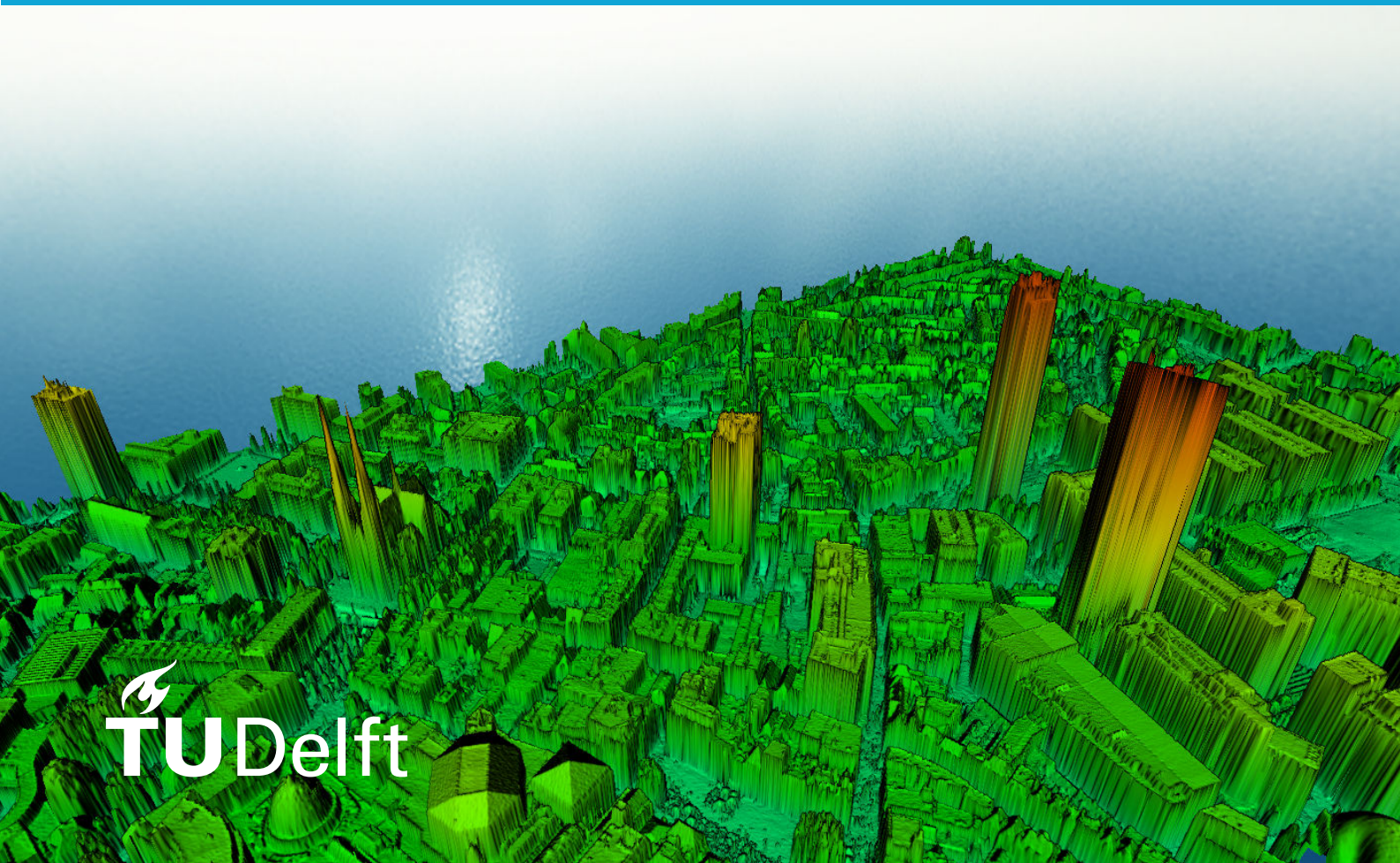
MSc thesis in Geomatics

Enhancing 3D Model for Urban Area with Neural Representations

Sitong Li

2024



MSc thesis in Geomatics

Enhancing 3D Model for Urban Area with Neural Representations

Sitong Li

July 2024

A thesis submitted to the Delft University of Technology in partial fulfillment of the requirements for the degree of Master of Science in Geomatics

Sitong Li: *Enhancing 3D Model for Urban Area with Neural Representations* (2024)
This work is licensed under a Creative Commons Attribution 4.0 International License.
To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/.

The work in this thesis was carried out in the:

3D geoinformation group
Delft University of Technology

Supervisors:	Dr. Ken Arroyo Ohori
	Nail Ibrahimli
Co-reader:	Dr. Azarakhsh Rafiee

Abstract

The accuracy and comprehensiveness of 3D city models have become increasingly important for applications like monitoring, sustainability evaluation, disaster management, and urban planning. However, creating accurate and complete 3D models is challenging. Traditional methods, such as photogrammetry and Light Detection and Ranging (LiDAR), often face issues like time-consuming processes and data gaps caused by occlusion. Additionally, these methods typically produce discrete models that fail to capture all the information from the original objects, limiting the ability to reconstruct small structures.

In this thesis, we explore the potential and characteristics of enhancing 3D models for urban areas using implicit neural representation. This method offers generalizability, ensures no void areas, and provides more detailed information when using multi-modal inputs. The datasets used for model training include Actueel Hoogtebestand Nederland 3 (AHN3), 2019 Luchtfoto Beeldmateriaal, 3D Basisregistratie Adressen en Gebouwen (BAG), and Basisregistratie Grootschalige Topografie (BGT). The city center of Eindhoven is used for training and testing, and the city center of Rotterdam serves as a test dataset.

The method involves learning location-dependent latent codes from raw point cloud and orthophoto, and adjusting the probability of space occupancy based on point clouds sampled from the 3D city model. A decoder is used to calculate the probability of existence for any point in the 3D space from the continuous field. Proper sampling allows us to derive a continuous Digital Surface Model (DSM) with unlimited resolution from the neural network.

The results show a high degree of accuracy: the generated DSM for the training area in Eindhoven demonstrated a median absolute error of 0.484 m overall. Accuracy for building areas was recorded at 0.798 m, and for terrain, it was 0.302 m. The model also displayed robust generalization capabilities, with an accuracy of 0.336 m in the Eindhoven test area and 0.23 meters in Rotterdam. Additionally, the model's ability to fill voids was confirmed through both visual inspection and quantitative evaluations. This research underscores the potential of implicit neural representation for generating detailed DSMs and effectively filling no-data voids.

Acknowledgements

This is always my favorite part when reading others' thesis since it serves both as an introduction and a conclusion to one's academic life. After 19 years of study, now it is my turn to write this section.

First and foremost, I want to thank my parents for always supporting and believing in me. Despite my less-than-stellar performance during my bachelor's studies, they continually encouraged me to preserve in study. Although I still feel uncertain about my future, much like when I graduated from Wuhan University two years ago, I believe that hard work will eventually pay off.

I would like to express my deepest gratitude to my supervisors, Ken Arroyo Ohori and Nail Ibrahimliand, for their guidance and companionship. They have always encouraged me during tough times and offered the best suggestions they could.

I also owe a lot debt of thanks to my friends. Without their support, I wouldn't have been able to complete my graduation project on time. My friends located all around the world, despite the time differences, have kept me company and warded off loneliness over the past seven months. Also, a special thanks to my company, Geodelta. Working with such talented engineers was an amazing experience. The valuable part-time job opportunity provided me with insights into the industry while allowing me to maintain a balance between work and study. The generous salary helped offset my high tuition fees, alleviating some of my financial stress.

Now, my time as an innocent and naive student is over. I shall step out of my shell and face reality. Goodbye and see you again ...

Contents

List of Figures

List of Tables

Acronyms

1. Introduction

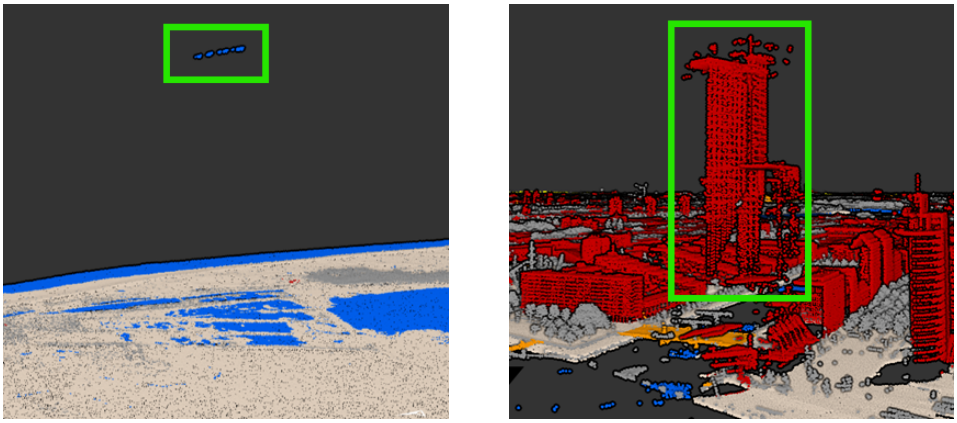
This chapter first gives an overview of the difference between explicit representation and implicit representation, and discusses the motivation for applying implicit representation in urban 3D model refinement. Section 1.2 reveals the research objective and scope by detailing several research questions. The chapter concludes with the organization of the thesis.

1.1. Background and motivation

Historically, the construction of 3D city models has relied heavily on techniques like photogrammetry and Light Detection and Ranging (LiDAR). Ground-breaking works by Ackermann [1999] and Vosselman and Maas [2010] laid the foundation for using aerial imagery and laser scanning for 3D reconstruction. These methods have been used for decades to achieve datasets of relatively high accuracy in urban modeling. The processing after data acquisition, such as data fusion and noise cleaning, is very costly. Nevertheless, the final reconstruction results are often unsatisfactory, frequently exhibiting problems like low point cloud density due to low reflectance of special planes, data voids due to occlusions from limited scanning angles, and noise that the algorithms cannot recognize as shown in Figure 1.1. For photogrammetry, the reconstruction result is often influenced by factors like weather conditions and terrain type, particularly in regions with sparse textures, for example, covered by clouds or water, resulting in discrepancies and missing data. For LiDAR sensors, the signals can be affected by disruptions like specular reflections, radar shadows, or delays on echo returns, especially over water surfaces or within deep mountain valleys. Such disturbances can also lead to incomplete data. Occlusion can also be an important reason for data missing. When it comes to the generation of DSM, the inconsistency in the raw point cloud will result in void gaps (Figure 1.2) and have negative impacts when it's used for further applications and analysis.

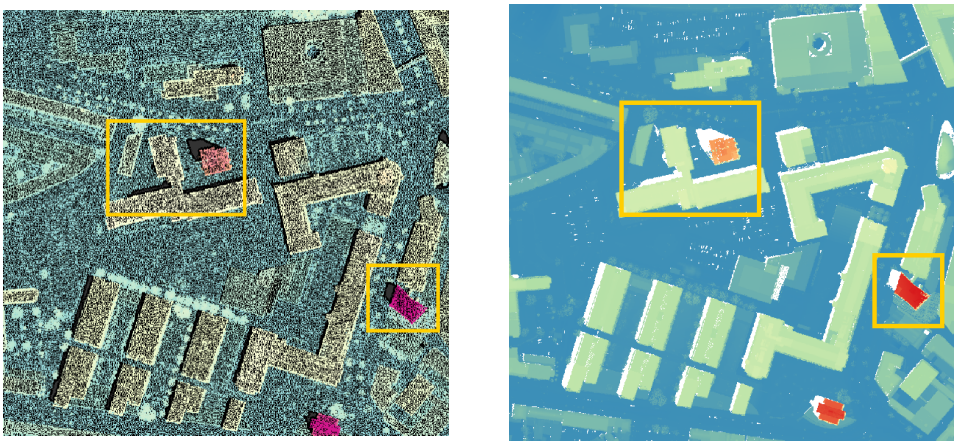
Before utilizing point cloud data for generating products such as Digital Terrain Model (DTM) and DSM, it is essential to clean the data. Additionally, the presence of voids in DSM can severely affect the analytical outcomes. Common approaches to enhance data quality include employing algorithms to remove airborne outliers from point clouds and performing interpolation to DTM and DSM to address minor voids. For larger gaps, integrating with reference data from other sources is more effective. However, these techniques generally rely on empirical, human-defined rules that require iterative adjustments tailored to the specific dataset.

Therefore, implicit neural representation is introduced. In Figure 1.3, a rabbit is shown in point cloud representation and implicit representation. The implicit representation is a continuous field that can store the signed distance from the current point to the surface or the occupancy status of any point within the bounding box. It has been used in many fields like 3D shape and scene reconstruction, signal processing, and fluid dynamics visualization.



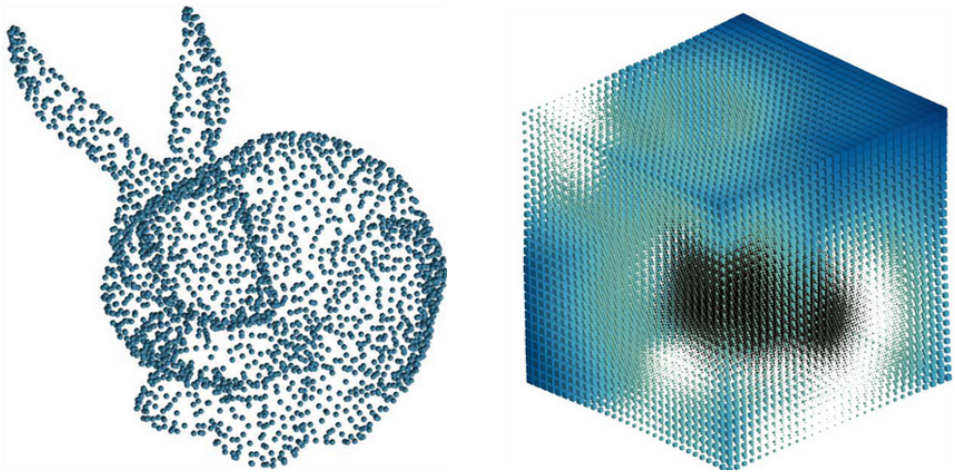
(a) Outlier points high in the air (b) Facade of the skyscraper is missing

Figure 1.1.: Errors in AHN4 point cloud



(a) Point cloud with missing data (b) Corresponding DSM with void area

Figure 1.2.: Gaps in raw point cloud data can cause void area in DSM



(a) Point cloud (b) Implicit representation

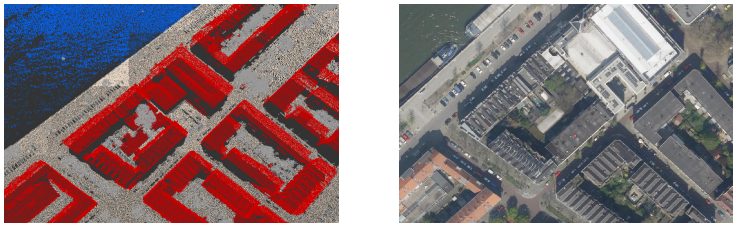
Figure 1.3.: Comparison between point cloud and implicit neural representation [Yadav, 2023]

The potential of neural representation in the field of Geomatics is both promising and yet to be fully explored. This approach excels at encoding complex, high-dimensional data into continuous, lower-dimensional forms, making it well-suited for spatial data. The network can learn from multi-modal input data, including point clouds from photogrammetry or LiDAR, and images, to output a continuous field while maintaining translation equivariance and transformation invariance.

This reconstruction approach allows for automated data processing that adapts dynamically to the complexities of the data. Unlike traditional methods that rely on predefined rules, implicit neural representations can learn to fill gaps and correct errors intrinsically, reduce the need for manual intervention. By directly learning from the data, these models are more robust to variations, thereby streamlining the reconstruction process.

Most of the current researches focus only on single-object reconstruction and room-scene reconstruction and there exist very few studies to apply implicit neural representation to real-scene geospatial data, particularly the open-source data from the Netherlands. Also, the lack of topology and continuity information in point cloud representation can be a big problem for the reconstruction of urban areas, especially for buildings with sharp edges and corners. In the study by Stucker et al. [2022], they apply the implicit neural network to point clouds reconstructed from WorldView-3 satellite images. Orthophotos are also incorporated into the implicit network, which can impart crucial insights on topology and continuity during the learning process, as indicated in Figure 1.4.

Given the continuous nature of the implicit field, the DSM can be generated with unlimited resolution both horizontally and vertically. This allows for precise calculations of the occupancy status at any given location. The continuous and generalizable nature of implicit neural representation has significant promise to improve the quality and utility of 3D models for urban areas. The generalizable nature of implicit neural representation effectively compensates for these missing parts, ensuring a complete and continuous surface representation in the final DSM.



(a) (b)

Figure 1.4.: Topology and continuity information available in image but not in point cloud

In this thesis, we extend the research to include the use of open-source datasets from the Netherlands, develop an alternative method for generating DSM, examine the model's generalization ability, and assess the effects of different training iteration times and thresholds on the generation outcomes.

1.2. Research objective and scope

The primary focus of this study is to evaluate the effectiveness of implicit neural representation for reconstructing urban scenes in the Netherlands, using data from various open-source databases. Input point clouds are sourced from AHN3, 3D city models from 3D Basisregistratie Adressen en Gebouwen (BAG), land use masks from Basisregistratie Grootschalige Topografie (BGT), and orthophotos from the Luchtfoto Beeldmateriaal. The output will be a DSM that is void-free and detailed. Its accuracy will be assessed by comparison with a DSM produced directly from point clouds (Figure 1.5).

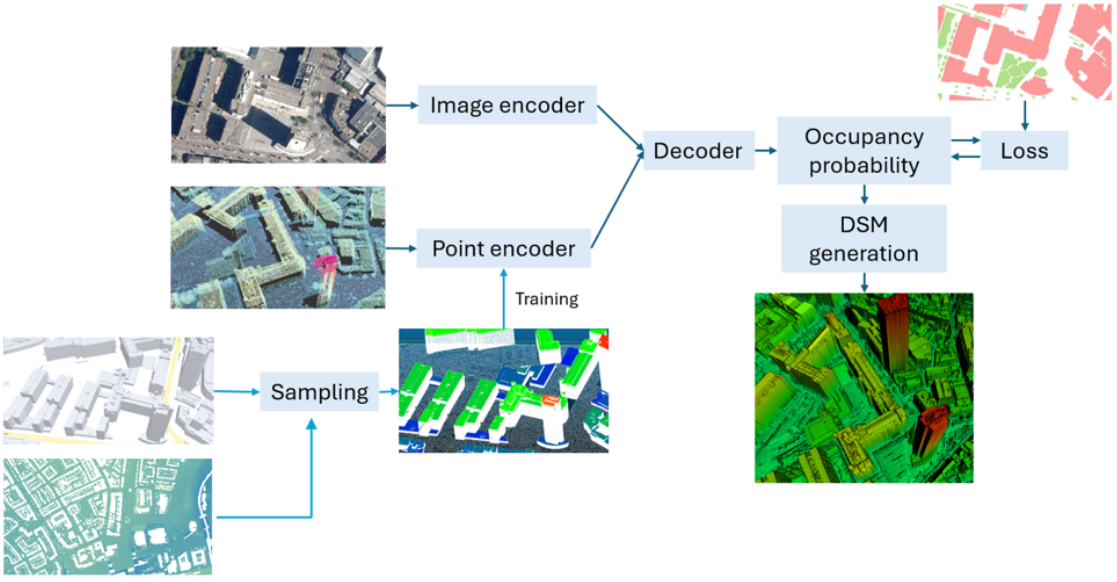


Figure 1.5.: Pipeline for Implicit Neural Representation

The main research question to be addressed is:

What are the characteristics of implicit neural representation when it's used for 3D real-scene urban area reconstruction?

It further subdivides into the following research sub-questions:

1. What process steps are needed to adapt current geospatial data to the network?
2. What is the geometric performance of implicit neural representation when applied to real-scene urban data reconstruction?
3. How effective is the generalizability of the implicit representation functions on AHN3 urban data?
4. Compared to traditional methods, what are the advantages and disadvantages of using the implicit neural representation for urban scene reconstruction with open-source datasets in the Netherlands?

1.3. Structure of the thesis

The thesis is organized as follows:

Chapter 2 reviews the related work on explicit representation, deep learning algorithms for reconstruction from point cloud and image, and recent studies on the deep learning reconstruction of large-scale real-scene data.

Chapter 3 outlines the methodology used in this study. It begins with data pre-processing, followed by a description of the network architecture and the principles behind the DSM generation algorithm from an implicit field. Finally, the chapter details the evaluation metrics employed, along with implementation details.

Chapter 4 presents the result of DSM generated using various generation methods, along with the test result of the generalizable ability of implicit representations. The limitations and application scope of implicit representation are also discussed.

Chapter 5 concludes this thesis by addressing the research questions and highlighting the contribution of the research. It also gives an outlook on future research.

Appendix A lists the web services commands used for assessing orthophotos and land use masks.

Appendix B contains the qualitative comparison between result with and without orthophoto as input.

Appendix C evaluates the reproducibility of the thesis.

2. Related work

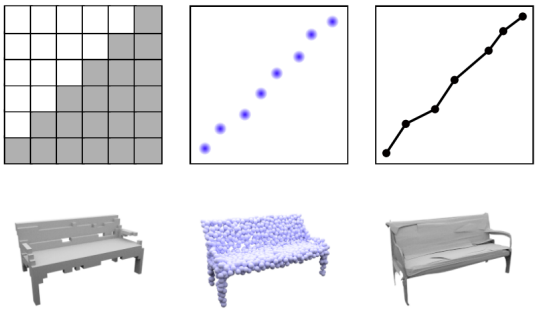
This chapter reviews various established and related methodologies pertinent to enhancing 3D models in urban environments, particularly through the innovative use of implicit neural representations. It begins with a discussion on explicit representations, including voxel, point cloud, and mesh methods, illustrating their constraints in capturing the intricate geometries typical of urban landscapes. As the focus shifts to more sophisticated techniques, deep learning for point cloud is explored for its transformative impact on processing unstructured spatial datasets. The chapter further investigates reconstruction from image methods. The next section focuses on adapting neural network methodologies to the complexities of real-world data, aiming to accurately scale and model urban environments in three dimensions. The final section critique conventional methodologies used to fill data gaps in surface models, underscoring the advantages of newer, more flexible approaches like implicit neural representations.

2.1. Explicit representations

The three most common ways for 3D data representation are voxel, point cloud and mesh as shown in Figure 2.1. However, each of them has certain limitations that hinder their ability to fully capture the geometric details of objects.

Voxel representation leads to complex implementations and existing data-adaptive algorithms like octree [Meagher, 1982] or multi-resolution shape reconstruction [Häne et al., 2017; Wang et al., 2018] are still limited to relatively small voxel grids (128^3 or 256^3). Moreover, this approach can cause Manhattan World bias because its grid-based structure aligns with a world composed of orthogonal planes and straight lines [Maturana and Scherer, 2015; Zishu et al., 2020] while the axis of large-scale real scenes is often affected by the curvature of the earth. Sampling errors are inevitable unless the resolution is smaller than the threshold calculated from Nyquist-Shannon sampling theorem [Shannon, 1949]. For implicit neural representation, the occupancy status of any position within the 3D space can be computed from the generated continuous field, eliminating resolution limitations in 3D model reconstruction. This method also allows the network to align with world coordinates, avoiding the need to define a separate coordinate system.

Point clouds, while capturing raw 3D data, do not contain the connectivity and topological structures. Consequently, additional postprocessing is needed to derive 3D geometries from the model, as indicated in studies by Li and Baciu [2021b]; Zhao et al. [2019]; He et al. [2019]. Additionally, point clouds struggle with effectively conveying the global shape of objects. This issue arises because point clouds represent 3D objects with no explicit information about the surface or volume these points belong to, which is crucial in applications requiring a holistic understanding of an object's form [Li and Baciu, 2021a]. These problems limit their direct applicability in computer graphics-related applications, such as shadow estimation. Furthermore, point clouds are often challenged by a limited number of points, which leads to a loss



(a) Voxel (b) Point (c) Mesh

Figure 2.1.: Three traditional representation methods [Mescheder et al., 2018]

of fine detail. Implicit neural representation allows the input dataset to include multi-modal data, such as images, enabling the model to learn topology and connectivity information. On top of that, this method supports the generation of various 3D models, including DSM [Stucker et al., 2022], DTM [Yao et al., 2024], and 3D city models [Chen et al., 2022] from the continuous field. This approach goes beyond point cloud generation, offering a more comprehensive understanding and representation of 3D objects for diverse applications.

Mesh representations are frequently used in discriminative 3D classification or segmentation tasks because they provide a structured and efficient way to represent 3D shapes and surfaces [Maturana and Scherer, 2015]. However, meshes always require class-specific templates (unique structural features) for each class for accurate modeling, which makes it difficult to adapt to objects with significantly different structures. Additionally, deforming these templates to fit specific instances can lead to issues like self-intersection [Reddy et al., 2022]. Implicit neural representation, on the other hand, represents 3D shapes as continuous functions, eliminating the requirement for predefined templates. This continuous representation also ensures that the surface defined through the level sets of functions is always continuous and intersection-free.

2.2. Deep learning for point cloud

The deep learning network for point cloud can be classified into multiview-based, voxel-based, and point-based methods based on its representation of data, and single object reconstruction and scene reconstruction based on the target size.

Multiview-based methods use structured 2D images to represent unorganized point clouds, but the projections process can lose geometric information([Lawin et al., 2017], [Su et al., 2015], [Boulch et al., 2018]), while for voxel-based methods like 3D CNNs, they are limited by the voxel resolution [Wu et al., 2015]. PointNet marks the beginning of point-based methods [Qi et al., 2017a]. The point-based methods can directly process each point in the point cloud preserving the most of geometry information and saving computation resources for pre-processing. However, its performance is bounded by not capturing local features from neighbouring points. Further studies have been focused on refining global feature aggregation

like Point Transformer [Engel et al., 2020], PointASNL [Yan et al., 2020] and RSNet [Huang et al., 2018] or other networks that use local feature aggregation like PointNet++ [Qi et al., 2017b], PointCNN [Li et al., 2018] and PointNGCNN [Lu et al., 2020].

This shift to point-based deep learning marked a departure from traditional geometry-based methods, paving the way for more sophisticated model generation techniques. Implicit neural representation is then applied showing great effectiveness and robustness for both single-object reconstruction and scene reconstruction. Implicit neural representation involves employing an encoder to learn features and patterns from discrete data, subsequently generating a continuous function that accurately represents the target object [Ran et al., 2022; Dai and Nießner, 2022]. Recent studies focus on enhancing the level of detail in 3D models. The works of [Park et al., 2019] on DeepSDF and [Chen and Zhang, 2019] have demonstrated significant advancements in capturing small details, which is particularly relevant for urban scene reconstruction where fine geometric features like building façades are crucial.

DeepSDF uses a continuous Signed Distance Function (SDF) to provide a novel and more flexible representation of 3D shapes. SDF associates an oriented plane (tangent plane) with each of the data points, then tracks the contour surface where the value equals zero, and finally, constructs the object surface from zero points (Figure 2.2). The deepSDF model learns the latent space of shapes, allowing it to generalize new, unseen shapes. This means the model can adapt to a variety of different 3D objects, not just specific instances from the training set. In addition, DeepSDF has the ability to learn continuous volumetric fields resulting in more accurate surface reconstructions. However, since it is directly mapping points to continuous SDF values through the neural network, its computational efficiency is relatively high.

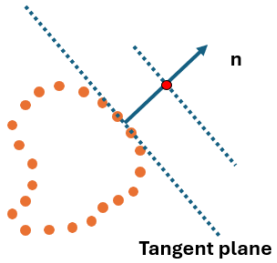


Figure 2.2.: Signed Distance Function

The Occupancy Networks introduced by [Mescheder et al., 2018] uses an indicator function instead of a signed distance function to reconstruct surfaces, reducing computational demands. It assigns an occupancy status to each sampled point, as defined in Equation 2.1. The points are sampled on the surface of the object and uniformly in the 3D bounding box of the object. The neural network then trains on a continuous function that predicts the occupancy probability based on the coordinates and features of any point. The network calculates mini-batch loss using sampled points, detailed in Equation 2.2, where B is the batch size, x_i is the i_{th} observation of current batch, K is the point number, o_{ij} is the real occupancy status at sampled point p_{ij}. Occupancy networks are proved to be good at recreating complex shapes and can work with different types of inputs like images, noisy point clouds, or simple 3D voxels. Its generative capabilities are tested through the results of unconditional generation.

$$[h]o : \mathbb{R}^3 \rightarrow \{0, 1\} \tag{2.1}$$

$$[h]\mathcal{L}_B(\theta) = \frac{1}{|B|}\sum_{i=1}^{|B|}\sum_{j=1}^{K}\mathcal{L}(f_\theta(p_{ij}, x_i), o_{ij}) \tag{2.2}$$

2.3. Reconstruction from image

The basic principle for image reconstruction is the coplanar condition equation. It assumes that the camera center, the image point and its corresponding object point are on the same line Figure 2.3.

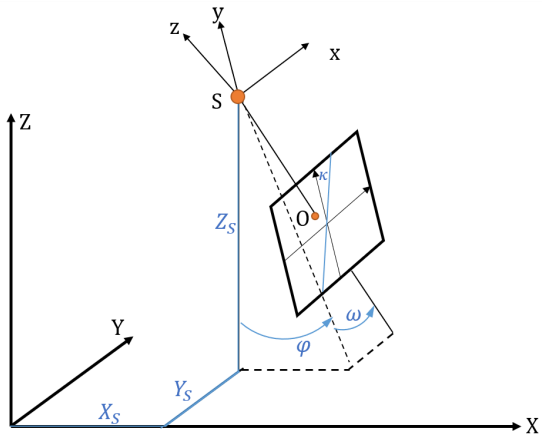


Figure 2.3.: Coplanar condition equation

The equation is formed with several elements. x' and y' are the coordinates of certain point on the image plane, X, Y, Z are the coordinates of the corresponding point in the world coordinate system, f is the camera focal length, X_S, Y_S, Z_S are the camera's optical center coordinates in the world coordinate system, also known as exterior orientation parameters, and a_i, b_i, c_i are the rotation matrix element that describes the camera coordinate system's orientation relative to the world coordinate system.

$$\begin{cases} x' = f \cdot \frac{a_1(X-X_S)+b_1(Y-Y_S)+c_1(Z-Z_S)}{a_3(X-X_S)+b_3(Y-Y_S)+c_3(Z-Z_S)} \\ y' = f \cdot \frac{a_2(X-X_S)+b_2(Y-Y_S)+c_2(Z-Z_S)}{a_3(X-X_S)+b_3(Y-Y_S)+c_3(Z-Z_S)} \end{cases} \tag{2.3}$$

When it comes to deep learning reconstruction from images, fully convolutional networks are used to preserve the spatial alignment between the image and the reconstruction result. Point representation [Li et al., 2020; Zhang et al., 2019], voxel representation [Xie et al., 2020; Choy et al., 2016] and mesh representation [Niemeyer et al., 2019; Pan et al., 2018] are also used as reconstruction output representations. Neural Radiance Fields (NeRF), introduced in [Mildenhall et al., 2020], reconstructs objects by creating novel views of complex scenes. It processes five dimensional inputs with three spatial coordinates (x, y, z) and two for viewing direction (θ, ϕ). Using volume rendering techniques, it calculates the volume density and view-dependent emitted radiance at these coordinates. The technique optimizes a continuous volumetric scene function based on a set of input views, refining the model for accurate 3D

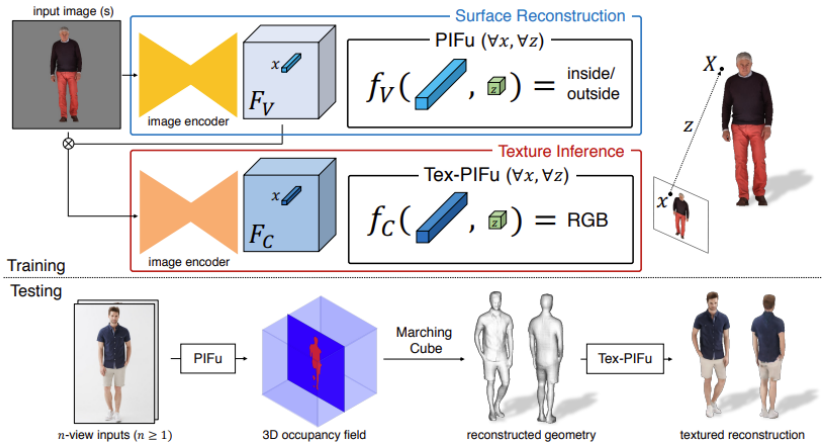


Figure 2.4.: Overview of PiFU pipeline [Saito et al., 2019]

rendering. However, NeRF suffers from low rendering speed, fixed illumination condition and limited generalization.

PiFU takes another way that learns an implicit function from the feature vector and z-depth for each pixel [Saito et al., 2019], as shown in Equation 2.4. Its pipeline is shown in Figure 2.4. Z-depth refers to the distance between the camera's optical center and the pixel's world coordinates. With the fully convolutional network architecture, local features of each pixel are aligned with its position in the world coordinate system. This strong correlation between image and 3D scene improves the accuracy of 3D shapes. The utility of implicit function allows for continuous and smooth surface reconstruction and enables unlimited resolution. What's more, its end-to-end training approach enables straightforward integration with other networks, minimizing the need for intermediate processing. In the thesis, the surface reconstruction part of the PiFU network is employed to provide guidance for reconstructing edges and corners.

$$f(F(x), z(X)) = s : s \in \mathbb{R}^3 \tag{2.4}$$

2.4. Reconstruction of real scene

Scalability has been a major issue in adapting current reconstruction methods to the real scenes. Most of the studies only focus on single-object reconstruction and neglect the coordinate alignment for real scenes. Convolutional occupancy networks [Peng et al., 2020] are among the first research efforts to consider scale adaptability. In the network, local features for each point are extracted using PointNet modified with local max pooling, projected orthographically to a canonical plane, and then decoded by Unet, a convolutional network. The utility of this convolutional network ensures translation equivalence, making it adaptable to different scales of reconstruction. The local feature extraction also enables the network to learn the repeating patterns or structural characteristics of the 3D scene. The network has

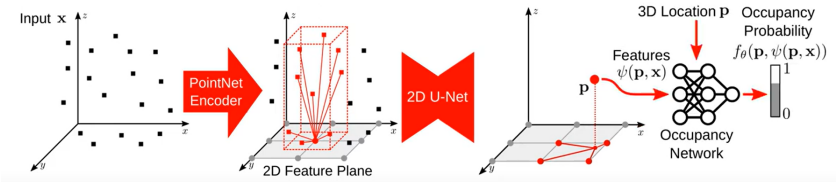


Figure 2.5.: Overview of convolutional occupancy networks [Peng et al., 2020]

been tested on single-object reconstruction and room scene reconstruction. The structure of convolutional occupancy networks is shown in Figure 2.5.

ImpliCity is a pioneering approach that applies implicit representation to real urban scenes. In this method, inputs are point clouds created through satellite imagery and orthophotos. Each point cloud and orthophoto is independently encoded yet mapped onto a unified canonical plane. The occupancy decoder then employs two distinct features as shown in Equation 2.5, unlike the single feature usage in Equation 2.1, to craft city models of greater detail. It is because that point cloud lacks inherent topology or connectivity, whereas the defining edges and corners in images are noticeable. In the equation, both $\psi(P, x)$ and $\xi(I, x)$ are coordinate-dependent, sharing the same x-y axis, facilitating training and testing on crops of data. For DSM generation, the process starts by generating a 3D grid, sized according to the desired DSM resolution and a predefined height range. Then, the vertical resolution is enhanced by subdividing the voxel at the current DSM height iteratively until the desired resolution is reached.

$$f_\theta(x, \psi(P, x), \xi(I, x)) \rightarrow \hat{o} \in [0, 1] \tag{2.5}$$

2.5. Traditional methods for DSM void filling

No-data values are unavoidable in 3D models reconstructed for both photogrammetry methods due to areas with weak or no texture in satellite images, and LiDAR technology due to signal interruption and occlusion, resulting in voids in the generated DSM. Various interpolation algorithms like Kriging [Matheron, 1963], Inverse Distance Weighted (IDW)[Shepard, 1968], Bilinear Interpolation[Diaz, 1957], and Bicubic Interpolation [Bhattacharyya, 1969] are used to fill these gaps based on the surrounding data. However, these methods are suitable only for small areas and perform poorly in water areas, with interpolated height values often differing significantly from their true values.

For larger void areas, external auxiliary data is required for data fusion. This method uses height values from other geographically aligned 3D models to create a complete dataset [Papasaika et al., 2009]. However, the accuracy of this approach largely depends on the reference data's accuracy. Additionally, differences in collection times, methodologies, and resolutions between the primary and auxiliary datasets can lead to data inconsistencies [Qiu et al., 2019; Zheng et al., 2016]. These inconsistencies can result in a lack of smooth transitions between the original and filled data, often creating a "step" effect, as shown in Figure 2.6.

Implicit neural representation, on the other hand, offers an innovative solution by learning a continuous function that represents the entire shape or surface. This method combines both lo-

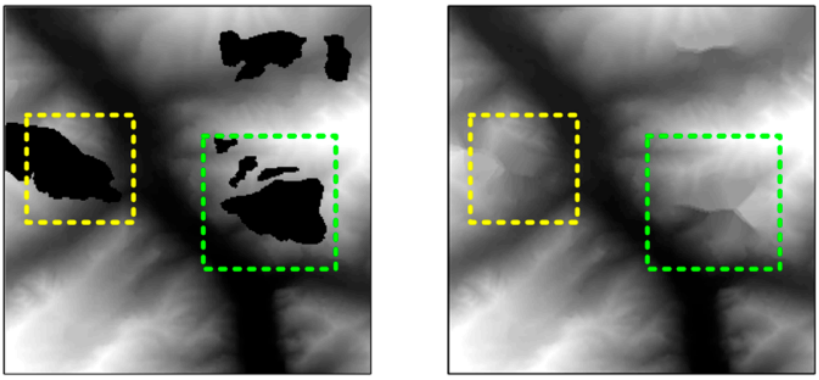


Figure 2.6.: ”Step” effect in data fusion result [Qiu et al., 2019]

cal and global understanding, allowing it to coherently infer missing parts and ensure smooth transitions between known and void data, thus avoiding artifacts like the ”step” effect.

3. Methodology

This chapter outlines the comprehensive workflow, including data acquisition and preprocessing, network architecture design, DSM generation, evaluation metrics, and implementation details.

3.1. Data Pre-processing

This part is to adapt the current open-source dataset in the Netherlands to the network of implicit neural representation. In the network, a raw point cloud and orthophoto are used as input, a query point cloud is used for back propagation, and masks of terrain, building, water, and vegetation are used during loss calculation.

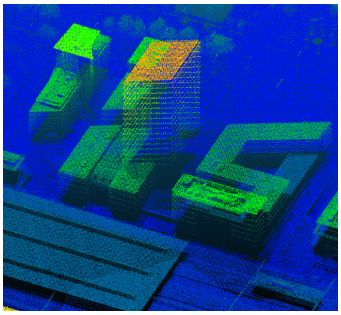
3.1.1. Input point cloud

The raw input point cloud used in this study is from the third Dutch national airborne laser scanning campaign, referred to as AHN3. This will be the source data for the DSM generation. In the traditional DSM generation method, for the target DSM grid, each cell will be assigned the elevation of the highest point falling within that grid. Then, void part will be filled using interpolation algorithms or other reference datasets. While for implicit neural representation, during training process, the encoder and decoder will takes in the point's coordinate and convert it into a occupancy probability between 0 and 1. A threshold is then used to set the division between occupied and unoccupied. Sampled point cloud and its occupancy status will be used for back propagation. During testing, the raw point cloud will go through the encoder and decoder, and output a continuous function. It can predict the occupancy probability at any given coordinate and DSM can be generated from it.

The DSM from AHN3 is not employed directly due to observed discrepancies between the DSM and the raw point cloud data, as shown in Figure 3.1. Reference DSM is generated from the input point cloud.

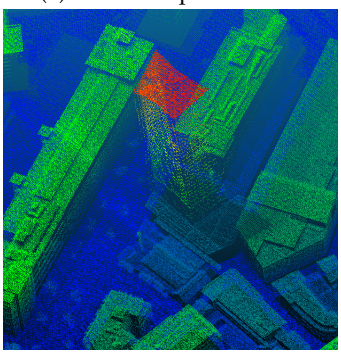
3.1.2. Query Point Cloud

The query point cloud, essential for occupancy status training, is sampled over the 3D city model (3DBAG), DTM, and uniformly around the space. The 3D BAG model at LOD 2.2, which includes detailed roof structures [Biljecki et al., 2016], is used in Figure 3.2. Ground points are sampled from the DTM in AHN3 because the 3D BAG lacks a ground surface representation. The sampled point cloud is further segmented into roof, facade, and ground surfaces, each of which undergoes specific sampling strategies that reflect their unique structural significance.

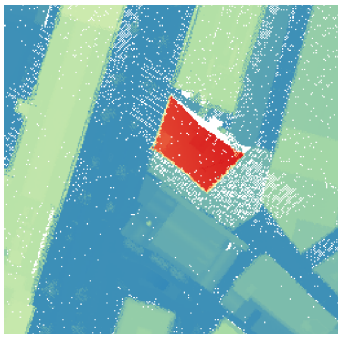


(a) Area 1 in point cloud

(b) Area 1 in DSM



(c) Area 2 in Point cloud



(d) Area 2 in DSM

Figure 3.1.: A lot of voids near towers in AHN3 DSM but not in the point cloud

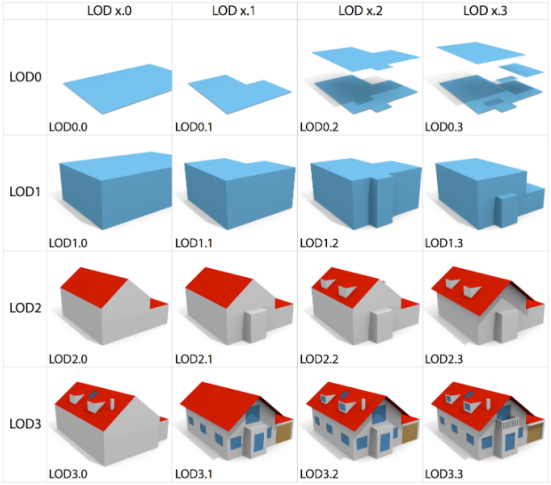


Figure 3.2.: Definition of LOD [Biljecki et al., 2016]

Different density param are assigned to each surface type to capture detailed geometric features appropriate to their complexity. For instance, roofs have denser sampling due to their intricate structures compared to ground surfaces. Additionally, terrain data is enriched with DTM, ensuring that the generated query point cloud can fully reflect the actual landscape.

The sampled point cloud is then divided into three categories based on elevation: above both DTM and DSM, above DTM but below DSM, and below DTM, as illustrated in Figure 3.3. This is because the vertical accuracy of 3D BAG has not been tested and it also contains outlier faces. The categorization helps mitigate the impact of noisy data during training.

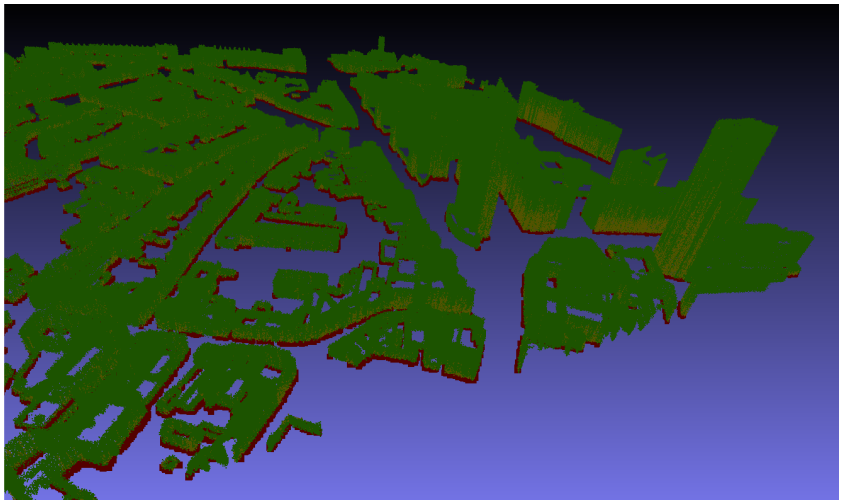


Figure 3.3.: Divided query point cloud

3.1.3. Orthophoto

Orthophotos serve as an additional input for the neural network, providing topological guidance by depicting straight lines and distinct corners. Orthophoto is image that have been geometrically corrected to eliminates distortions caused by central projection and the camera angle. This correction ensures a uniform scale across the entire image, allowing for accurate representation of the earth's surface and precise extraction of features directly from the image. Additionally, orthophoto should share the same axis as the point cloud to align features extracted from both the point encoder and the image encoder. To enhance efficiency, the images are converted to black and white, minimizing computational resource demands.

3.1.4. Masks

Masks for buildings, water, vegetation, and ground are used. These masks should be in raster format to facilitate easy class assignment to the query point cloud. During loss calculation, different classes of points are assigned varying weights, with ground and buildings given higher weights compared to vegetation and water.

(a) Accuracy for some building polygon is low

Feature	Value
pand [3]	
identificatie	0599100000617974
(Derived)	
(Actions)	
gid	1716638
bouwjaar	1935
identificatie	0599100000617974
pandstatus	Pand in gebruik
geconstateerd	false
documentdatum	1/1/1975
documentnummer	01/3744/72
voorkomenidentificatie	1
begindatumtijdvakgeldigheid	1/1/1975 00:00:00 (UTC)
einddatumtijdvakgeldigheid	10/16/2015 00:00:00 (UTC)
tijdstipregistratie	8/27/2010 18:39:30 (UTC)
eindregistratie	10/16/2015 17:33:29 (UTC)
tijdstipinactief	NULL
tijdstipregistratielv	8/27/2010 19:01:23 (UTC)
tijdstipeindregistratielv	10/16/2015 18:00:02 (UTC)
tijdstipinactieflv	NULL
tijdstipnietbaglv	NULL
aanduidingrecordinactief	false
geom_valid	true

(b) Registration time is not complete and hard to understand

(c) Underground part is also included in the BAG

Figure 3.4.: Problems with existing land use building data

Directly applying building masks from BAG or BGT presents challenges due to issues with temporal resolution, accuracy, and data definition, as indicated in Figure 3.4. Since AHN3 point cloud has already been classified into mutiple classes, in this study, we directly use building type of points from it to create building masks by projecting onto the x-y plane and converting to a binary format.

3.2. Network architecture

The network consists of three main parts: the point encoder, the image encoder, and the occupancy prediction decoder. It's overview can be seen in Figure 1.5. Input point cloud and orthophoto works as two separate inputs of the network, the extracted feature is combined together in the decoder and the final output is a continuous function. Query point cloud and masks are used for back propagation and loss calculation.

The point encoder uses the method described in Section 2.4, originated from the work of Peng et al. [2020]. The image encoder employs the technique outlined in Section 2.3, originated from the work of Saito et al. [2019]. The occupancy prediction decoder follows the method presented in Section 2.4, originated from the work of Stucker et al. [2022].

3.2.1. Point encoder

The structure of the point encoder is shown in Figure 3.5. In point feature extraction, local max pooling helps refine the feature map by preserving local features and exploring the self-similarity of the building structure. Applying U-Net to the feature plane ensures the network's translation equivalence, allowing it to adapt to data at different scales.

The first part of the network is feature expansion from point feature (x, y, z) to a 64-dimension vector with a linear layer. The data then passes through a ResNet [He et al., 2016] block and the feature dimension is deduced to 32.

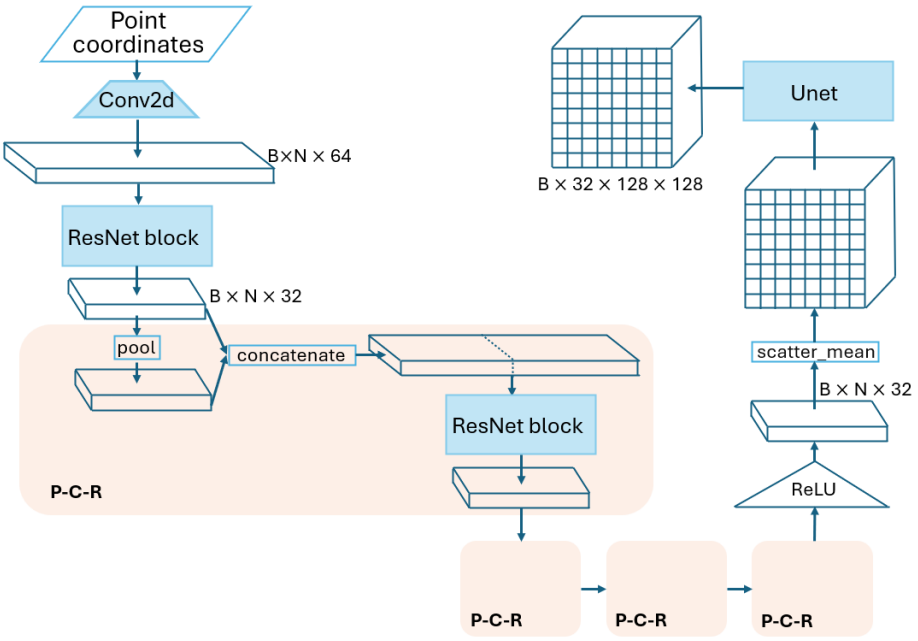


Figure 3.5.: Pipeline of point encoder

Then, a "pool-concatenate-ResNet" process is repeated four times, as shown in the P-C-R block in Figure 3.5. The output from the local max pool is concatenated with the previous output (before pooling) and passes to another ResNet block.

The output tensor $B \times N \times 32$ from the repeated steps is then passed through a ReLU activation function, introducing non-linearity to the model, which helps in learning complex patterns. An index is then assigned to each point based on whether their x,y coordinates fall within a specified grid cell. The features are then aggregated to create a structured format using an appropriate function that averages these features within each grid cell based on their indices. This allows the transformation of scattered data into an organized multi-dimensional tensor, facilitating more structured analysis or further processing. It generates a 128×128 feature plane, where each cell in the grid represents the mean feature value of the points that fall into that grid cell. This step effectively downsamples and summarizes the feature space into a more manageable form.

U-Net from [Huang, 2017] is then used for the point feature plane. The U-Net is designed to conserve translation equivariance, meaning the output from the U-Net will change in a predictable way if the input is translated. This property helps to highlight the relative positioning of features within the data, making the network compatible with different scales.

3.2.2. Image encoder

The structure of image encoder is shown in Figure 3.6.

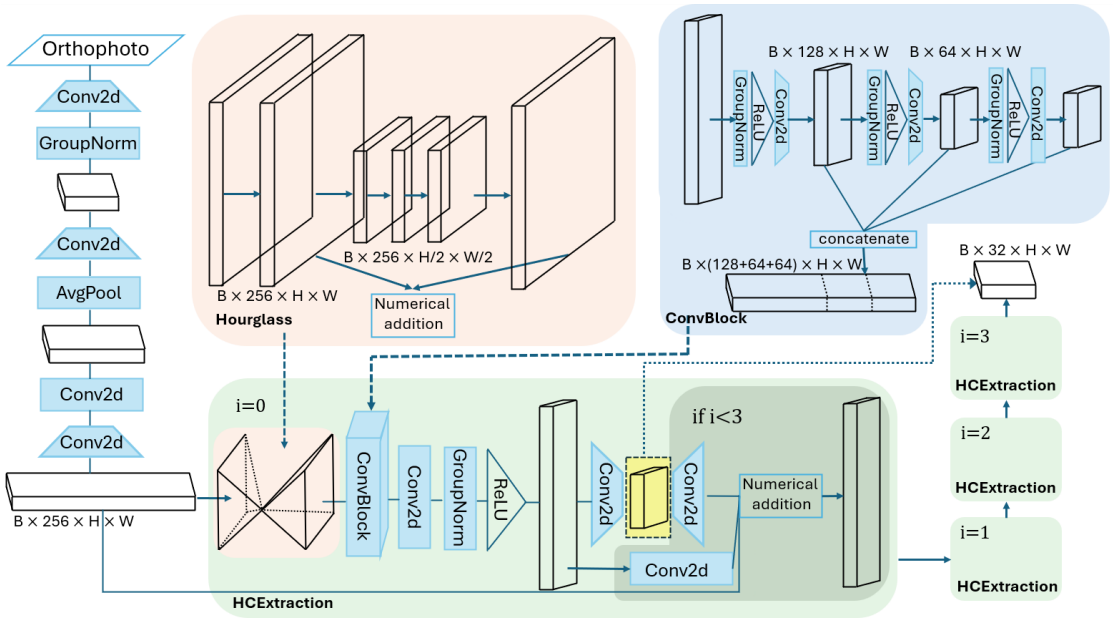


Figure 3.6.: Pipeline of image encoder

Hourglass network structure is used for extracting image features at various scales and integrating them to enhance the feature representation. The hourglass design, characterized by repeated downsampling and subsequent upsampling, allows the model to capture both high-resolution details and broader contextual information from the images. This ability to operate at multiple resolutions is crucial for tasks requiring fine-grained detail as well as global understanding. Other than that, the encoder also includes a specially designed convolutional layer block that concatenates outputs from different stages of the network. This concatenation process yields a composite feature map that encompasses a diverse range of learned features at multiple levels of abstraction. By integrating these features, the network can leverage both low-level textural information and high-level semantic details, enhancing the model's ability to interpret and analyze complex image data.

The training starts with feature expansion, employing group normalization and average pooling. Group normalization helps to stabilize the training by normalizing the inputs across subsets of channels, while average pooling enhances translation equivalence by summarizing the features within each pooling window.

The $B \times 256 \times H \times W$ tensor is then put into an iterative process as shown in the HCExtraction block in Figure 3.6. It starts with a two-layer hourglass network. Hourglass networks are used for capturing information at various scales, especially when the image number is low. The tensor is passed through convolutional layers that downsample it, followed by layers that upsample the resolution back to the original height and width. Outputs before and after the hourglass network are merged using numerical addition, as shown in the Hourglass block.

After the hourglass network, the tensor undergoes group normalization followed by Rectified Linear Unit (ReLU) activation and passes through a ConvBlock composed of several convolutional layers with varied param. This sequence is repeated three times, each producing outputs at different stages. The outputs from these three stages are concatenated. After

concatenation, the combined feature map is processed through another set of convolutional layers and normalization, ending with an activation function to produce a final tensor of size $B \times 256 \times H \times W$.

The tensor then passes to two separate convolution layers. The first one decreases the feature dimension to 32; this is the output for the final iteration. Then the decreased dimension is processed back to 256. The second one keeps its dimension at 256. The outputs of these two pathways are then combined through numerical addition, and the result is recycled as input for the next iteration of the process. The final output is a tensor of size $B \times 32 \times H \times W$.

3.2.3. Occupancy prediction decoder

Figure 3.7 illustrates the decoder pipeline. It combines the features extracted from the image and the point clouds together since they are coordinate-aligned. This integration helps the network to learn complementary information from both sources, addressing the lack of topology and continuity in the point cloud.

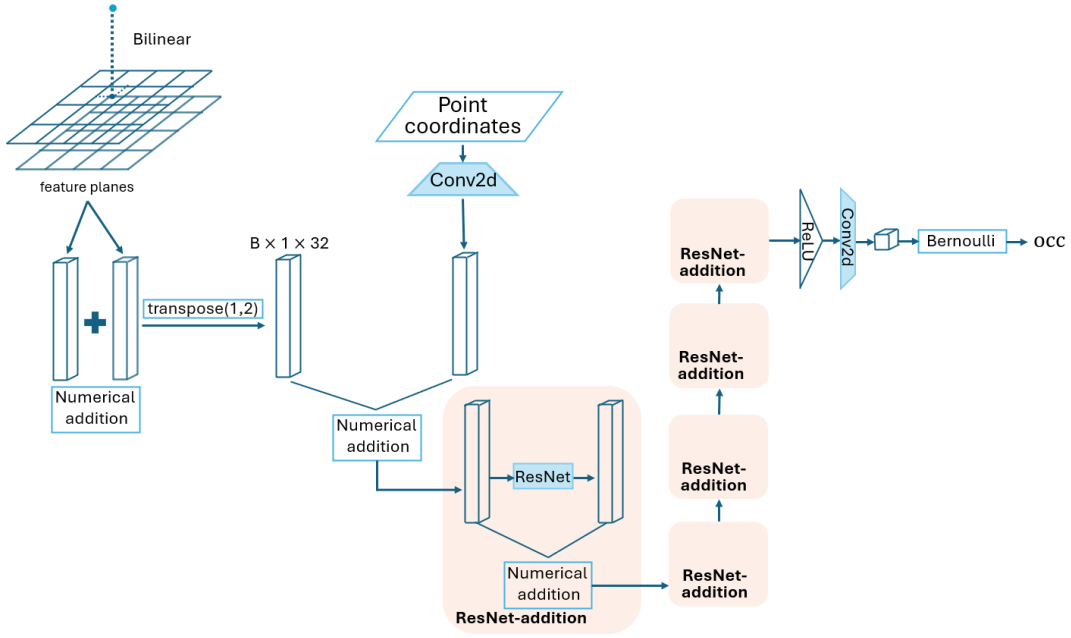


Figure 3.7.: Pipeline of occupancy prediction decoder

The output from the point encoder and image encoder are aligned feature planes, which ensure spatial correlation of features from both modalities. For each query point, unique feature vectors are computed by performing bilinear interpolation on the feature planes. The two feature vectors encoded from the point cloud and image are then added numerically. Additionally, the feature is enhanced by adding the expanded feature from point coordinates, incorporating spatial context into the feature set. The combined feature then passes through a series of ResNet-addition blocks for five times, in which the output before and after the ResNet

is added numerically. After the iteration, the output is activated using a ReLU activator, followed by a 2D convolutional layer that reduces the feature dimensions to a single-dimensional output. Finally, the feature is passed through a Bernoulli distribution layer, which maps the 1-dimensional feature to a probability value between 0 and 1.

3.2.4. Loss function

In this study, a binary cross-entropy loss is used. It is often applied to binary classification tasks which is suitable for the probabilitic outputs of the network. For query points sampled within a 3D city model, DTM, and uniformly across the bounding box, their predicted occupancy probability $\hat{o}$ from the network and the actual occupancy status o assigned during sampling are used for loss calculation. The true occupancy status o_i is sourced from established 3D models relevant to the training area. During data generation, a structured 3D grid of query points is sampled hierarchically, as detailed in the next section. The loss is computed as follows:

$$\mathcal{L}(\hat{o}, o) = \sum_i [o_i \cdot \log(\hat{o}_i) + (1 - o_i) \cdot \log(1 - \hat{o}_i)]$$

Here, the first term handles the loss when the true occupancy status o_i is 1. If the prediction $\hat{o}_i$ is close to 1, the logarithm of a number close to 1 is 0, resulting in a low loss for that sample point. However, if $\hat{o}_i$ is near to 0, the logarithm of a small number becomes a large negative number, leading to a high loss, which reflects the poor prediction. When the true occupancy o_i is 0, the second term comes to use. Since the predicted occupancy probability is among $[0, 1]$, $1 - \hat{o}_i$ is used to measure how close it is to the correct occupancy. The losses are summed together to give a collective measure of the accuracy. The binary cross-entropy loss is used for its high efficiency.

3.3. DSM generation

The final step is to extract the 3D model. The encoder-decoder network can transform the input point cloud and orthophoto into an implicit function. Using the calculated features and the query point coordinates, it can determine the occupancy probability at each query point. In theory, the resulting model can achieve unlimited resolution because the implicit function is continuous. The challenge lies in identifying the appropriate query points and finding an effective method to convert probability into an actual 3D geospatial product.

In the context of DSM, it is often referred to as a ”2.5D” model. A 2.5D model differs from a comprehensive 3D model that can represent multiple layers, as it is restrict to representing a single elevation value for each point on the x-y plane (Figure 3.8). Additionally, the elevation is selected as the highest point within a specific cell. This unique limitation makes the generation much easier since more constraints can be added other than only using an occupancy probability threshold.

Given that the network’s output is a continuous field, it enables the determination of the occupancy probability at any point. When it comes to DSM generation, this capability allows generating DSM at any horizontal and vertical resolution. A 3D grid is created for occupancy prediction at the desired horizontal resolution but considerably low vertical resolution. After

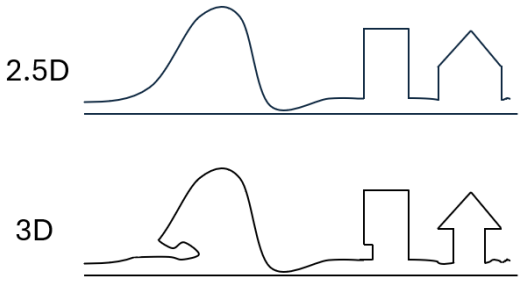


Figure 3.8.: 2.5 D and 3D model

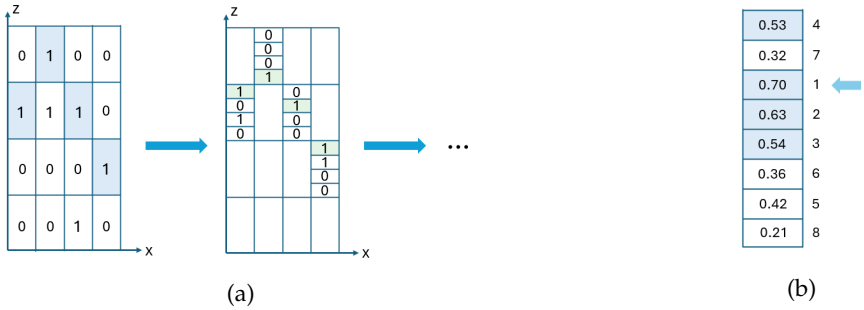


Figure 3.9.: Two DSM generation methods

that, iterations of processing will be done to refine the height value. For each iteration, the occupancy status at the center point of a voxel is predicted. If the predicted occupancy probability exceeds a predefined threshold, the voxel's status is marked as occupied. If the status of the current voxel is occupied and it is the highest of its x-y column in the z direction, then the voxel is divided into four smaller voxels for further iteration (Figure 3.9a). The process is repeated until the desired vertical resolution is achieved [Stucker et al., 2022].

Another method is top-n probability method that use the occupancy probability. For voxels in the same column, their occupancy probability will be sorted. Only the one with a probability larger than the threshold, ranking among the top-n and the highest in the z-direction, will be put into the next interaction, as shown in Figure 3.9b. The top-n probability method ensures that only the most probable voxels are considered in each iteration, leading to a more precise height estimation. In addition, in the highest cell method, a voxel is split if it is marked as occupied and is the highest in its x-y column. This can result in inaccurate height measurements, especially in regions with noisy data or sudden height changes. The top-n probability method, however, minimizes the impact of outliers by focusing on voxels with the highest probabilities, thereby reducing the likelihood of erroneous height values. In this study, n is set to 2.

Furthermore, to further minimize outliers, a correction procedure is added. For each cell, it extracts the center value and its 3×3 neighborhood, calculates absolute differences, and checks for conditions where no neighbors have the same height or all differences exceed 2. If the condition is met, it applies K-Means clustering to separate the remaining neighbors into two clusters, representing roof and ground elevations. Depending on the center value, it selects the appropriate cluster, calculates its mean, and assigns this mean value to the center position

if inconsistencies are detected, printing diagnostic information as needed. It is conducted only to cells higher than the 0.8 multiply max elevation to save computation resources. This is because that we found on the building edge and roof of skyscrapers has the highest residual. But for building edges, it is hard to decide with elevation to follow so the correction is only conducted to cells on the skyscraper roof.

Furthermore, several post-processing algorithms for the DSM have been developed. The first algorithm involves using building masks to merge DSMs of varying accuracy and adjust potential height changes to roofs. This approach is beneficial because DSMs created with different models, methods, and thresholds excel at representing various parts of the urban area, and combining them can result in a more accurate overall DSM. Another algorithm focuses on removing water areas from the generated DSM. The primary method involves extracting a water mask from the AHN3 DSM using an area threshold and then removing these areas from the generated DSM. This is necessary because water levels can fluctuate due to tides or rainfall during data collection, making it challenging to achieve consistent and reliable measurements.

3.4. Evaluation metrics

3.4.1. Evaluation on generated DSM

The evaluation of the generated DSM focuses on its accuracy and generation capability. The residual is calculated by pixel-by-pixel subtraction of the generated DSM from the reference DSM generated from the point cloud. The following parameters are calculated for both the complete dataset and for individual surface types. In the equations, z_i is estimated elevation for current x-y position and $\hat{z}_i$ is the "real" elevation from the reference DSM.

1. Maximum and minimum residual
2. Mean Absolute Error (MAE)

$$\text{MedAE} = \text{median}(|z_i - \hat{z}_i|) \tag{3.1}$$

3. Root Mean Square Error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(z_i - \hat{z}_i)^2} \tag{3.2}$$

4. Median value
5. Median Absolute Error (MedAE)

$$\text{MedAE} = \text{median}(|z_i - \hat{z}_i|) \tag{3.3}$$

6. Normalized Median Absolute Deviation (NMAD)

$$\text{NMAD} = 1.4826 \times \text{median}\left(|z_i - \hat{z}_i - \text{median}\left(|z_i - \hat{z}_i|\right)|\right) \tag{3.4}$$

In the equation, 1.4826 is used as a scaling factor to convert the Median Absolute Deviation (MAD) into an estimator for the standard deviation under the assumption that the residuals follow a normal distribution.

7. Number of pixels

3.4.2. Generalization evaluation

The generalization capabilities of the model are tested by applying the model to new, unseen data. Furthermore, visual and quantitative evaluations are done for no-data filling. The quantitative evaluation involves intentionally omitting points from various areas and locations within the input point cloud, as illustrated in Figure 3.10. This simulates the loss of data that might occur during data collection. A comparison is then performed between the generated parts of the DSM and the original DSM.

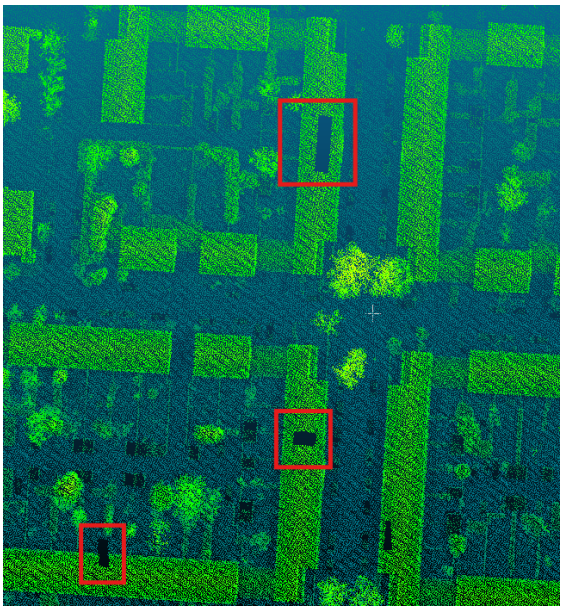


Figure 3.10.: Omitting points from input point cloud

3.5. Implementation Details

3.5.1. Data Pre-processing

AHN3 is collected by LiDAR technology from 2014 to 2019 during the leaf-off season, boasts a density of approximately 10 points per square meter, and contains over 700 billion points covering the entirety of the Netherlands [Kissling et al., 2023]. Its height accuracy is 20 cm and planetary accuracy is 23 cm according to AHN products overview. The classification of the point cloud includes six categories: Never Classified, Unclassified, Ground, Building, Water, and Reserved (such as bridges). Point Data Abstraction Library (PDAL) is used to merge and

crop .laz files. The corresponding DSM and building mask are created using the software GlobalMapper (Figure 3.11). The resolution of the generated DSM is set to 0.5 m, consistent with the DSM resolution in AHN3. The parameter "grid 'No Data' distance criteria" is set to 0, ensuring that the software performs no interpolation. The input point cloud for DSM generation can be filtered based on its classification. The DSM generated with building points only is used as a building mask. The orthophotos are sourced from the 2019 Luchtfoto Beeldmateriaal dataset,

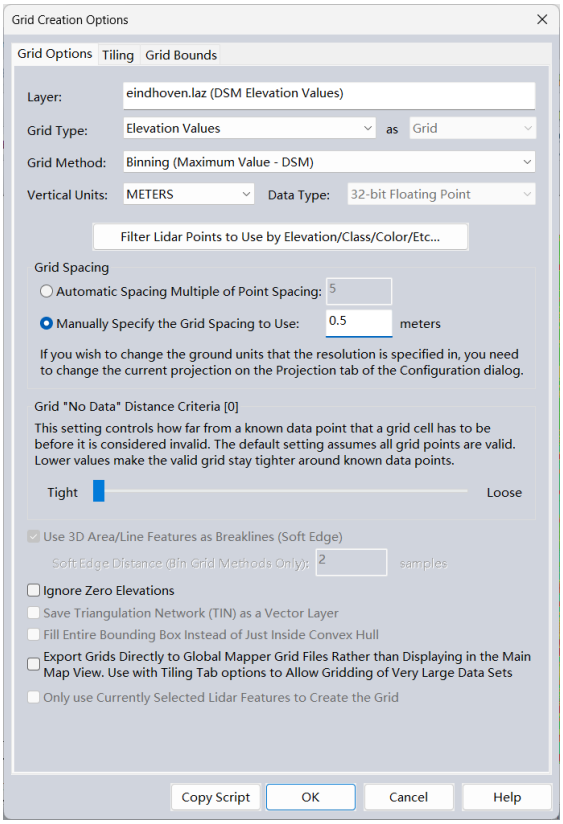


Figure 3.11.: Create elevation grid with GlobalMapper

which boasts a resolution of 25 cm. These can be accessed via Luchtfoto data room or by employing the Web Map Service (WMS) command as detailed in Appendix A.

The .obj file of the 3DBAG is used for sampling.

Masks for water and plants are available for download from PDOK or via the Application Programming Interface (API), as listed in Appendix A. The vector datasets need to be converted to raster format using the Geospatial Data Abstraction Library (GDAL) command. As demonstrated in Section 3.1.4, building masks need to be generated using classified building type points in GlobalMapper. If the software is not available, an alternative option is to use Web Feature Service (WFS) to download the dataset from BAG or API from BGT. Notably, there exist differences between buildings in BAG and BGT. Once a building is permitted, it will be registered in BAG, while only if it has been physically built, will it be included in BGT. In the inclusion of the geometry, the BAG geometry concerns the top view of a building, while the BGT geometry is at ground level [Geonovum, 2020; Odijk and Brnobic, 2021].

It is important to note that there are data quantity limitations with web services. Thus, it is highly recommended to divide the study area into smaller sections, retrieve the datasets incrementally, and subsequently merge them.

3.5.2. Study Area

The city centers of Eindhoven and Rotterdam have been selected as the study areas. In Eindhoven, the study area measures 2043 m in height and 1103 m in width, covering an area of 2.25 km². For Rotterdam, the study area measures 665.03 m in height and 1139.12 m in width, encompassing an area of 0.76 km².

During data preprocessing, the study area is clipped into 600 m $\times$ 600 m chunks. The test area of Eindhoven are divided into 8 chunks. During training, chunk 0,1,2,3,5 are used for training and chunk 4 is used for validation. Chunk 6 and chunk 7 are used for testing. The distribution is shown in Figure 3.12. The training and testing area includes a variety of buildings such as

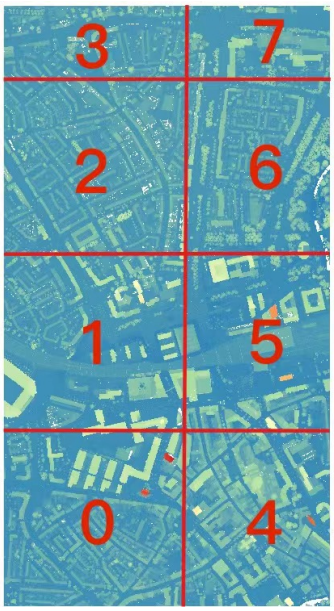
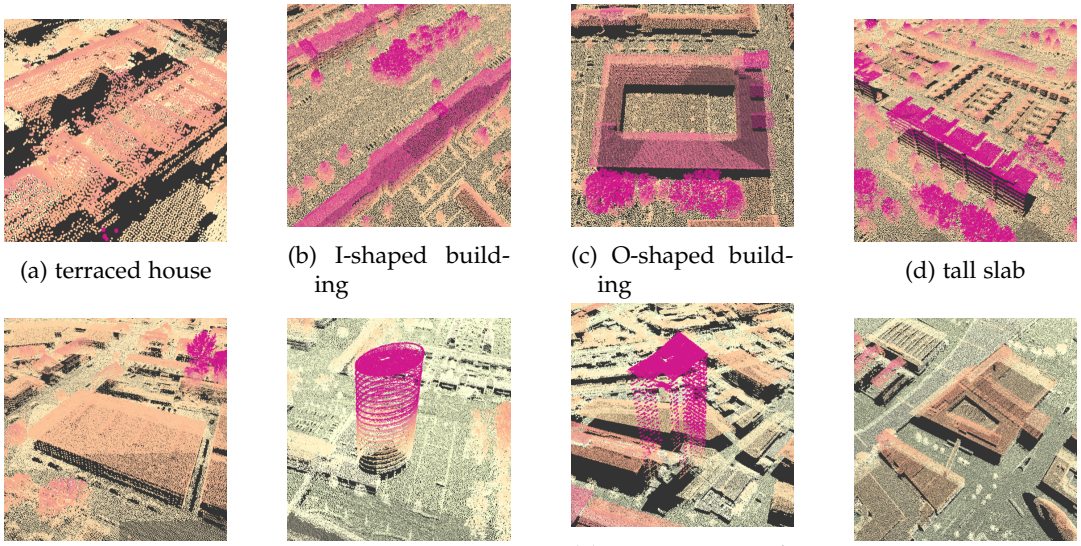


Figure 3.12.: Chunks of test area in Eindhoven

terraced houses, I-shaped buildings, O-shaped buildings, tall slabs, thin slabs, towers, towers with podiums, and complex structures, as illustrated in Figure 3.13.

3.5.3. Network hyperparameters

During the training and testing phases, data is segmented into patches measuring 112 m$\times$112 m. Each patch should encompass an entire building to ensure complete learning of the structure.



(a) terraced house (b) I-shaped building (c) O-shaped building (d) tall slab

(e) thin slab (f) tower (g) tower with podium (h) complex building

Figure 3.13.: Different type of buildings within the study area

For the image encoder, the feature dimension is established at 32, and the plane resolution for maximum scattering is configured at 128. The U-Net is designed with a depth of 5 to ensure the receptive field fully covers the target area. The feature dimension for the image encoder remains at 32. The architecture includes four hourglass-structured Multi-Layer Perceptron (MLP)s, each with a depth of 2. In the decoder, bilinear sampling is used, and the feature size is maintained at 32.

3.5.4. Hardware, libraries, and software

The implementation was conducted on a Tesla T4 GPU with RAM of 15 GB. The main libraries and software are listed as followed:

1. **PyTorch** [Paszke et al., 2019]. The network is built upon the PyTroch framework.
2. **PDAL**. The library is used for point cloud preprocessing, including merging and clipping to desired area.
3. **Rasterio**. The library is used during DSM generation mainly for handling raster data reading, writing, and geospatial transformations.
4. **Global Mapper**. The software is used for reference DSM and building mask generation and is also used as the 3D viewer.
5. **QGIS**. The software is used for vector-raster data conversion with GDAL based tools and also serves as the 2D viewer.

4. Results and discussion

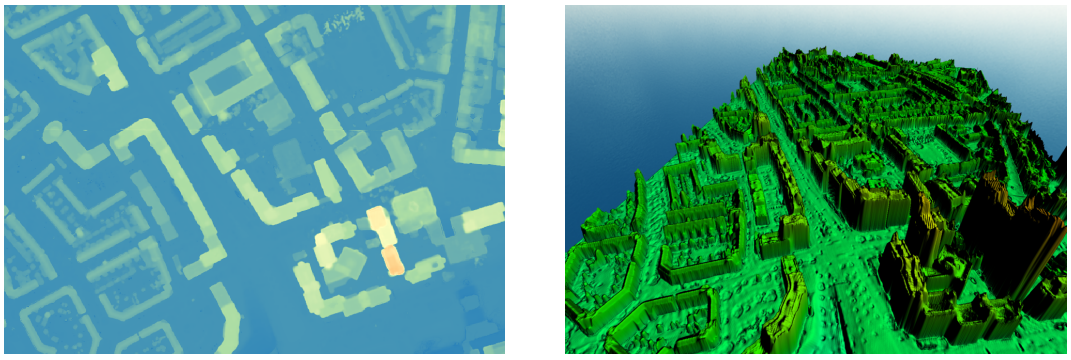
This chapter presents the outcomes derived from various DSM generation methods from the implicit occupancy field. It also encompasses the assessment of the model's generalization abilities and its proficiency in filling no-data value. The results section concludes with an analysis of the effects of iteration time and various generation thresholds. The discussion section elaborates on the limitations and application scope.

4.1. Generation result of two different methods

4.1.1. Highest cell method

Visualization of generated DSM

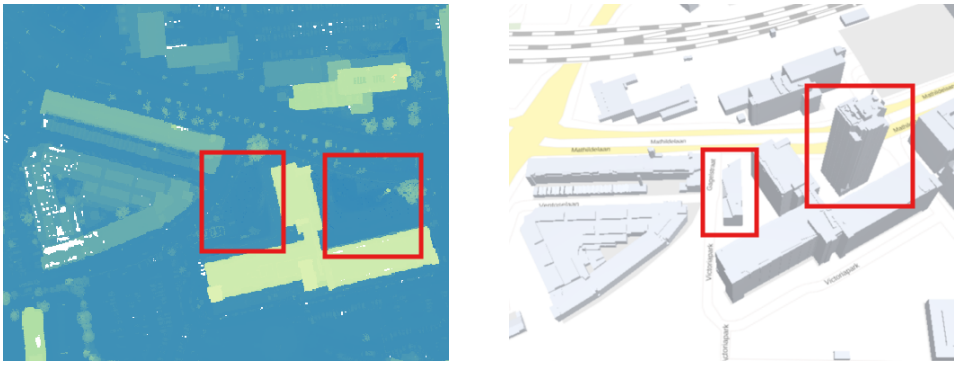
Using the highest grid generation method described in Section 3.3, a DSM with fine geometric details is produced. The model is the result of 5200 iterations, with a generation threshold set at 0.7. Figure 4.1 illustrates the generated DSM in both 2D and 3D views. It is evident that some roof details are particularly well-defined, with even small features being visible. A grid pattern appears on flat roofs because height estimation is performed cell by cell, leading to minor height differences between cells.



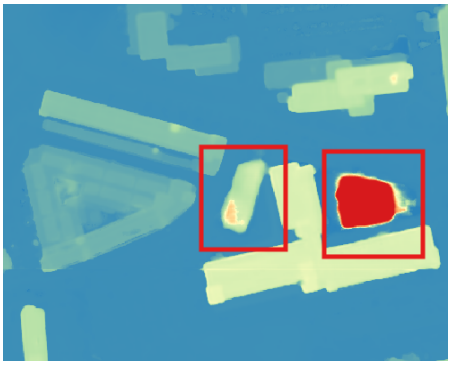
(a) (b)

Figure 4.1.: Generated DSM in 2D and 3D view

Additionally, due to the time difference, buildings are not identical in the input point cloud and the query point cloud sampled from 3DBAG. As shown in Figure 4.2, two buildings were not constructed when the AHN3 dataset was captured, but they already exist in the 3DBAG. This results in outlier buildings in the generated DSM.



(a) AHN3 dataset (b) 3DBAG dataset



(c) Generated DSM

Figure 4.2.: Differences between input point cloud and query point cloud result in outliers in the generated DSM

Evaluation

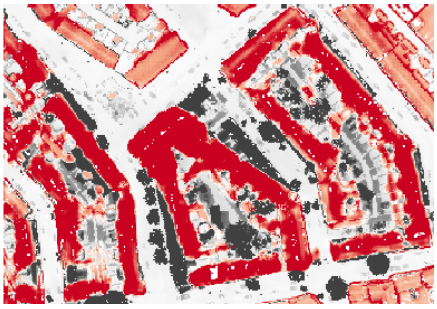
Table 4.1 is the evaluation result for the generated DSM:

Type	MedAE[m]	Median[m]	MAE[m]	RMSE[m]	NMAD[m]	Pixels
Overall	1.2828	0.2093	2.7408	6.6293	2.1450	17209506
Building	2.1471	-1.9780	3.6042	6.4280	6.2400	3529615
Vegetation	0.5515	0.4728	3.5183	8.6550	0.5023	718374
Terrain	0.3573	0.3118	2.0316	5.3886	0.2266	6862569
Terrain_no_vegetation	0.3497	0.2985	1.8928	5.2770	0.2114	4937153

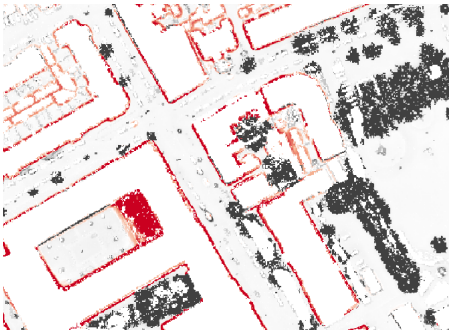
Table 4.1.: Evaluation of result from the highest cell method

Although MAE and RMSE are commonly used for accuracy evaluation, they do not fully reflect the true accuracy of the generated dataset in this study. The high MAE and RMSE values can be caused by the previously mentioned differences between AHN3 and 3DBAG (Figure 4.3a). The model's difficulty in handling sudden height changes also has contribute to it, as illustrated in Figure 3.9a and Figure 4.3b. The model aims to encode the entire scene into a continuous

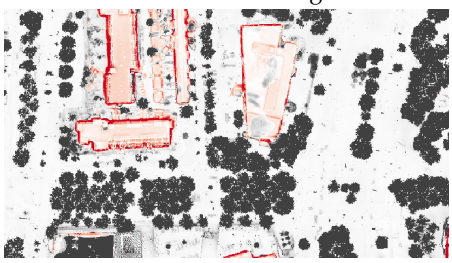
function, which can lead to confusion when there are abrupt height changes in buildings, resulting in structures that appear larger than they should be. Most importantly, the high residuals are intensified by the aggressive elimination of vegetation (Figure 4.3c). This occurs because the terrain surface of the sampled point cloud is devoid of vegetation, having been derived from a DTM. Additionally, a low weight is allocated to plant and water areas during the loss calculation to diminish their influence on building generation. Consequently, the model is trained to eliminate vegetation, resulting in a DSM with virtually no plant presence. Instead, median and MedAE provide a better assessment.



(a) High residual caused by differences in AHN3 and 3DBAG



(b) High residual caused by building edges



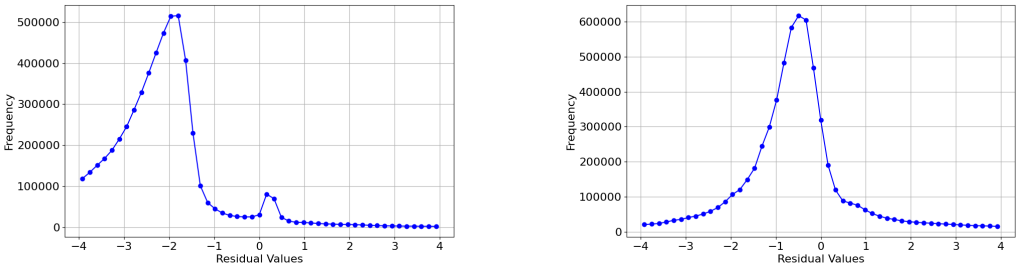
(c) High residual caused by elimination of vegetation

Figure 4.3.: Potential causes for the high error

As we can see from Table 4.1, the RMSE of vegetation (8.6550 m) is the highest among all categories, suggesting some large errors, which is consistent with what we assumed before. Its low MedAE and median error is because that the vegetation mask covers larger area than just trees and bushes that show obvious height change. Terrain has the lowest errors across almost all metrics compared to other categories, with MedAE of 0.3573 m, indicating high accuracy on this surface type. Both median error and NMAD are low, confirming consistent accuracy in terrain measurements. However, for building surfaces, it shows a notably higher MedAE (2.1471 m) and MAE (3.6042 m), which indicate challenges in accurately measuring in built-up areas. The Median error is -1.9780 m, which is negative, indicating a systematic underestimation by the model in building areas. Despite the high errors, the NMAD is markedly high at 6.2400 m, pointing to significant variability in the accuracy across this category. Further training is needed to reduce the error.

4.1.2. Top-2 probability method

The top-2 probability method can generate better roof heights by effectively narrowing down the candidate voxels to those with the highest probabilities. This results in a dramatic improvement in the accuracy of buildings. Figure 4.10 shows the distribution of residuals for building pixels, where the x-axis represents the residuals and the y-axis represents the number of pixels. The residual for most pixels has improved from around -1.9 meters to approximately -0.5 meters. Figure 4.11 compares the building residual maps of the highest method and the top-2 probability method.



(a) Distribution of building residual of the highest cell method

(b) Distribution of building residual of the top-2 probability method

Figure 4.4.: The residual for most building pixels generated by top-2 probability method is much lower than the highest cell method

4.1.3. Combined DSM result

Although the top-2 probability method can generate buildings with high accuracy, its ground generation results are poor. To address this, a merging algorithm is used, as introduced in Section 3.3.

The evaluation result for the merged DSM is shown in Table 4.2. As presented, the overall metrics provide a summary across all categories. The MedAE is relatively low at 0.4843 m, indicating generally reliable accuracy. A very low median error of 0.1882 m points to minimal bias across the entire dataset. The RMSE of 6.2917 m, while higher than the MAE, indicates the presence of some large errors which significantly impact the mean of the squared errors, as we explained ealier.

The MedAE for buildings shows a great improvement from 2.1471 m in Table 4.1 to 0.7983 m, and the median is reduced from -1.9780 m to -0.4930 m, suggesting relatively accurate reconstruction. This improve that the top-n probability method can effectively pick the most probable voxels and generate a more precise height estimation. For terrain and vegetation areas, the MedAE and median are all smaller than 0.5 m. According to the general rule that data accuracy should be at least 1/1000th of the map scale's denominator, a MedAE of 0.8 meters translates to suitability for map scales where 1 map unit corresponds to 800 ground units. Thus, a map scale of approximately 1:1000 can be made based on the generated DSM.

(a) Building residual of the highest cell method

(b) Building residual of the top-2 probability method

Figure 4.5.: The residual maps show better accuracy for top-2 probability method

Type	MedAE[m]	Median[m]	MAE[m]	RMSE[m]	NMAD[m]	Pixels
Overall	0.4843	0.1882	2.2267	6.2917	1.0183	17209506
Building	0.7983	-0.4930	1.9334	4.0296	2.1755	4535085
Vegetation	0.4801	0.4085	3.3444	8.5057	0.4615	718374
Terrain	0.3016	0.2636	2.7728	8.6806	0.2045	6230403
Terrain_no_vegetation	0.2900	0.2543	2.6568	8.1074	0.1816	4328152

Table 4.2.: Evaluation of the combined DSM

4.2. Generalization ability

To evaluate the model's performance on new, unseen data, it was applied to an untrained part of Eindhoven and a part of the city center of Rotterdam. Both generation results show nearly the same accuracy as the trained areas, with buildings displaying fine details.

The generation result for Eindhoven is shown in Figure 4.6. Table 4.3 is the quantitative evaluation result.

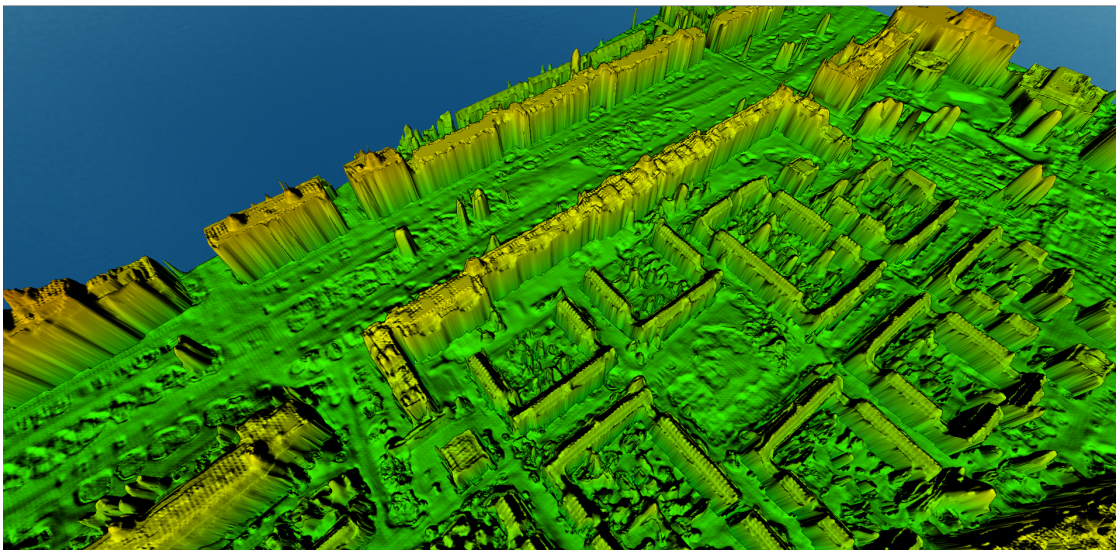


Figure 4.6.: Generation result for test area in Eindhoven

Type	MedAE[m]	Median[m]	MAE[m]	RMSE[m]	NMAD[m]	Pixels
Overall	0.3364	0.2281	2.4177	5.2011	0.4988	6066614
Building	0.7137	-0.4039	1.9222	4.1182	1.9224	822627
Vegetation	0.2777	0.2609	3.9082	7.3576	0.4117	915725
Terrain	0.2747	0.2663	2.7092	5.7631	0.4073	3697489
Terrain_no_vegetation	0.2999	0.2885	2.0520	4.5059	0.2693	1702021

Table 4.3.: Evaluation of the result for the Eindhoven test area

The terrain and topography of the study area in Rotterdam differ significantly from the training set, including large areas of water, multiple tall buildings, and generally lower elevation compared to Eindhoven. The final generated DSM still maintains a comparable level of accuracy. Figure 4.7 is the generation result of the test area in Rotterdam and Table 4.4 is the quantitative evaluation.

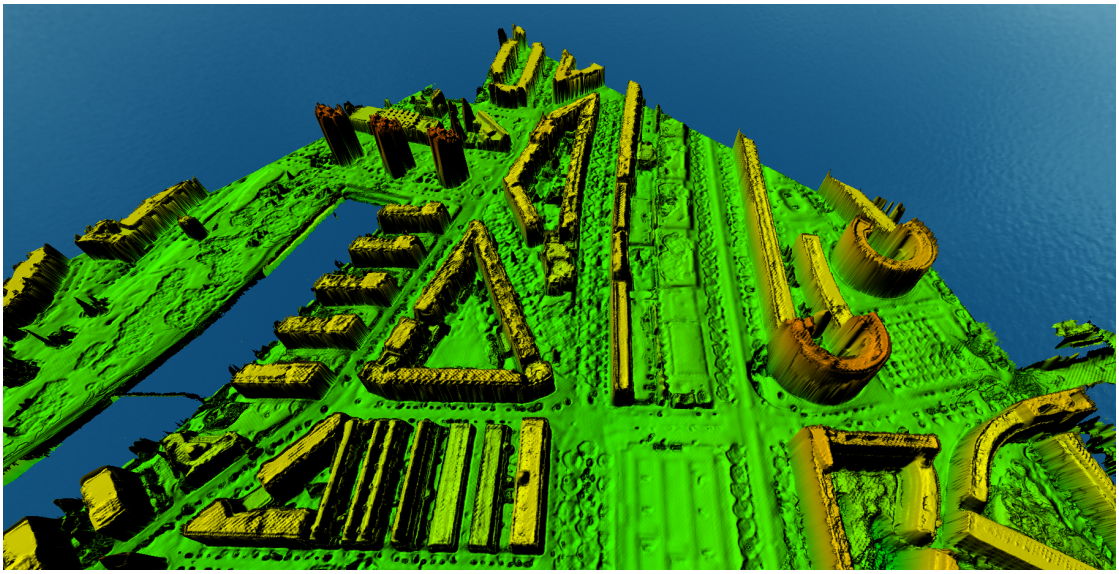


Figure 4.7.: Generation result for test area in Rotterdam

Type	MedAE[m]	Median[m]	MAE[m]	RMSE[m]	NMAD[m]	Pixels
Overall	0.2304	0.0000	1.4960	3.9871	0.3416	11942849
Building	0.8582	-0.6525	1.6034	3.3675	2.4043	1123399
Vegetation	0.4043	0.3548	3.1362	6.3861	0.3088	940520
Terrain	0.0000	0.0000	1.3113	3.8818	0.0000	8617636
Terrain_no_vegetation	0.0000	0.0000	0.8120	2.8603	0.0000	6730096

Table 4.4.: Evaluation of the result for the Rotterdam test area

The evaluation of the generated DSM in the Eindhoven and Rotterdam test areas shows the model's strong generalization capabilities across varied environments. In Eindhoven, the model demonstrates reasonable accuracy with a MedAE ranging from 0.2747 m in terrain to 0.7137 m in buildings, suggesting moderate precision in estimating building heights and terrain features. However, the relatively high RMSE across all categories, especially in vegetation (7.3576 m) and terrain (5.7631 m), indicates the presence of significant outliers or errors.

Rotterdam's test results further affirm the model's adaptability, particularly in handling diverse urban topographies, including areas with tall buildings and expansive water features. Notably, the DSM achieves exceptional accuracy in terrain measurements, with zero MedAE and median error, which is significantly better compared to Eindhoven. This improvement suggests that the model is well-suited to Rotterdam's simpler terrain features or better aligns

with the training data's characteristics. The overall and vegetation metrics in Rotterdam show a marked improvement from Eindhoven, indicating the model's robustness in adapting to different environmental conditions and complexities.

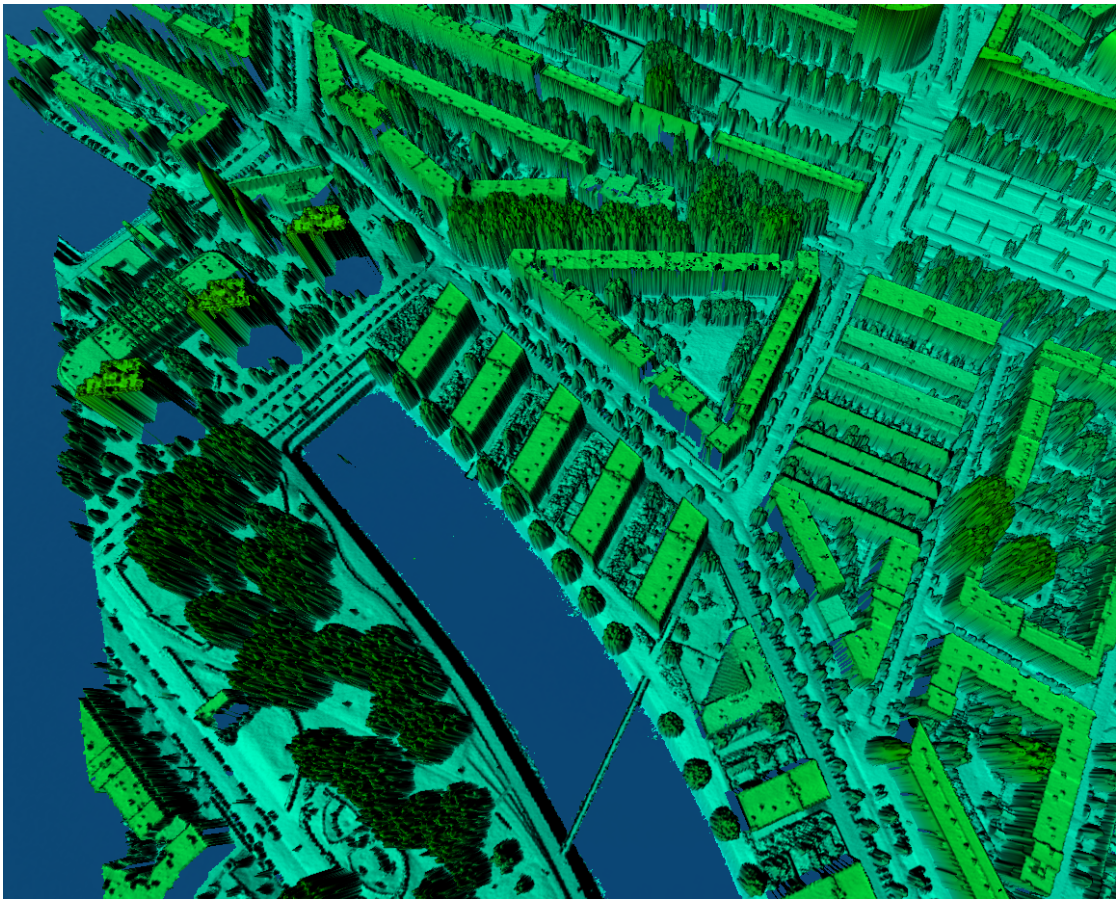
4.3. DSM void filling ability

Another important feature of implicit neural representation is its ability to generate a continuous field even when there are missing parts in the input data. In DSM generation, this results in a continuous surface, effectively addressing the problem of void filling without manual intervention, thus saving significant time and resources. As presented in Figure 4.8, the generated DSM can fill the void in input data seamlessly. To quantitatively evaluate the model's performance with missing data, we intentionally remove points in a specific area, like in Figure 4.9, and then compare it with the original DSM. A total of 21 polygons were used and Table 4.5 is the comparison result.

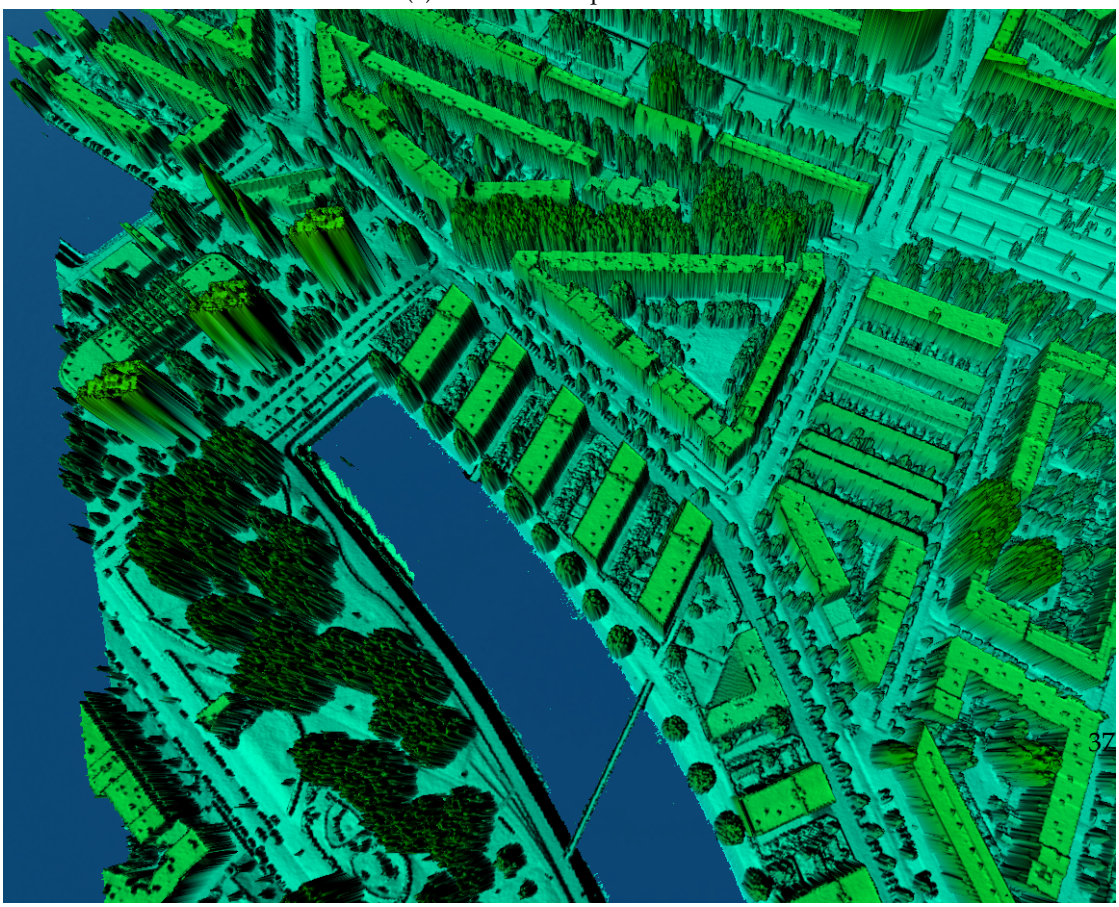
ID	Area [m^2]	MAE [m]	RMSE [m]	MedAE [m]	Median [m]	NMAD [m]	Pixels
1	9.986	0.151	0.239	0.070	-0.070	0.104	280
2	15.457	0.205	0.307	0.105	0.086	0.155	368
3	40.386	0.146	0.228	0.061	0.000	0.091	1100
4	7.032	0.284	0.391	0.258	-0.258	0.383	171
5	7.572	0.058	0.081	0.065	0.000	0.096	204
6	19.829	3.799	5.033	3.411	0.000	5.057	391
7	6.908	0.927	1.191	0.653	0.653	0.968	144
8	4.815	0.635	1.178	0.232	0.000	0.345	112
9	1.846	0.082	0.101	0.029	0.108	0.043	44
10	13.631	0.222	0.319	0.135	0.000	0.201	275
11	19.409	0.210	0.306	0.156	-0.156	0.232	522
12	25.898	5.039	7.693	1.990	0.000	2.950	576
13	15.392	1.322	2.172	0.208	-0.145	0.308	377
14	31.031	0.357	0.499	0.226	0.049	0.335	608
15	40.914	0.562	0.852	0.358	0.000	0.531	819
16	26.872	2.707	3.858	1.630	-0.384	2.417	540
17	17.220	0.190	0.255	0.178	0.000	0.263	374
18	76.187	1.295	2.119	0.416	0.000	0.617	1950
19	46.613	0.906	1.940	0.112	0.000	0.167	1302
20	39.662	0.838	1.032	0.549	-0.955	0.814	720
21	88.886	1.908	2.092	0.336	-2.189	0.498	1664

Table 4.5.: Evaluation of polygons with internal points removed

For most of the void-filling results, the MAE and RMSE are smaller than 0.4 meters, indicating precise filling of voids. However, discrepancies arise with larger voids or complex structures. The accuracy strongly correlates with the area and location of the polygons. Polygons with low errors are typically located in the middle of roofs and most have an area smaller than 20 m^2, except for Polygon 3. Despite its larger area of 40.386 m^2, Polygon 3 is situated on a gable roof and maintains the structure of this specific roof type. In contrast, Polygons 18 (76.187 m^2) and Polygon 19 (46.613 m^2) disrupt the building structure by cutting them in half, confusing the model and resulting in incorrect surface regeneration (Figure 4.10).



(a) DSM from the point cloud



(b) Void areas filled with the generated DSM

Figure 4.8.: Void filling in existing DSM using implicit neural representation

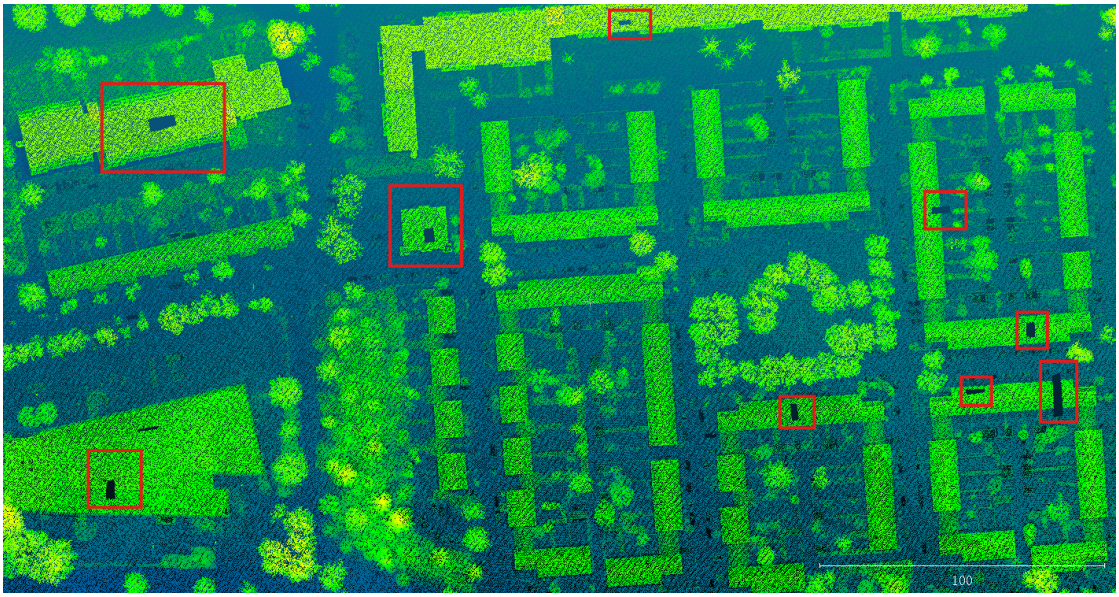
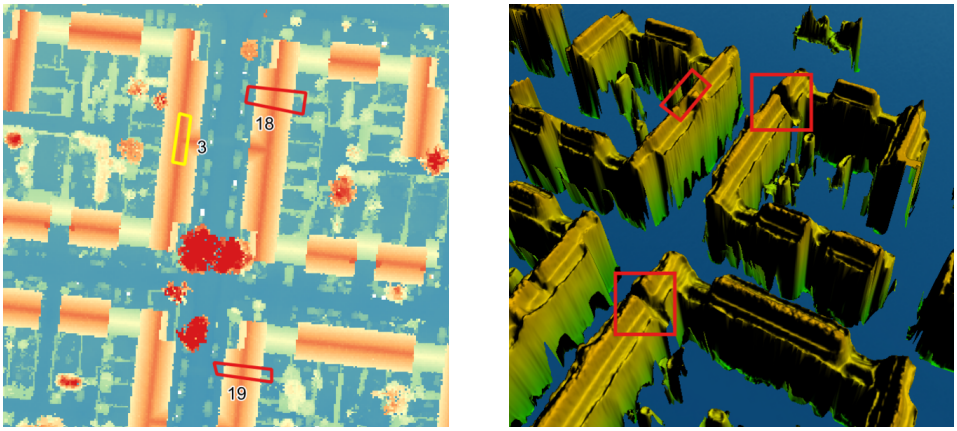


Figure 4.9.: Intentionally created holes in input point cloud



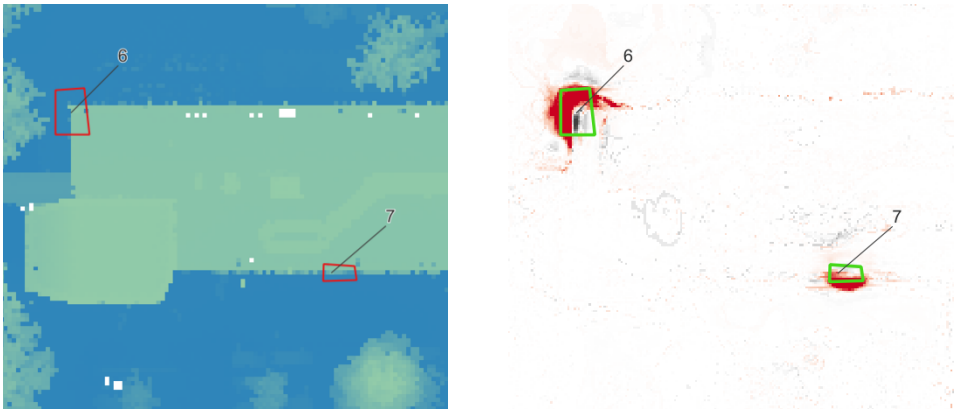
(a) Location of the polygon No. 3, 18 and 19

(b) Generation result with points removed within the polygons

Figure 4.10.: For gable roofs, maintaining structural continuity is more important than the area of the missing data

Further analysis shows the model's limitations at the edges of buildings or where there are abrupt changes in height. For instance, Polygons 6 and 7, although relatively small in area (19.829 m^2 and 6.908 m^2 respectively), exhibit higher errors (MAE of 3.799 m and 0.927 m, respectively). These polygons are located at the boundaries of buildings where partial ground points are removed, leading to substantial discrepancies in height representation and consequently higher residuals. This underscores a notable aspect of the implicit representation model's weakness in dealing with sudden topographical changes, which can lead to inaccura-

cies in DSM generation (Figure 4.11).



(a) Location of Polygon 6 and 7 on the DSM

(b) Residual comparison with and without the hole

Figure 4.11.: Generation performance for polygons at the edge of buildings is sub-optimal

4.4. Effect of iteration number and threshold

4.4.1. Iteration number

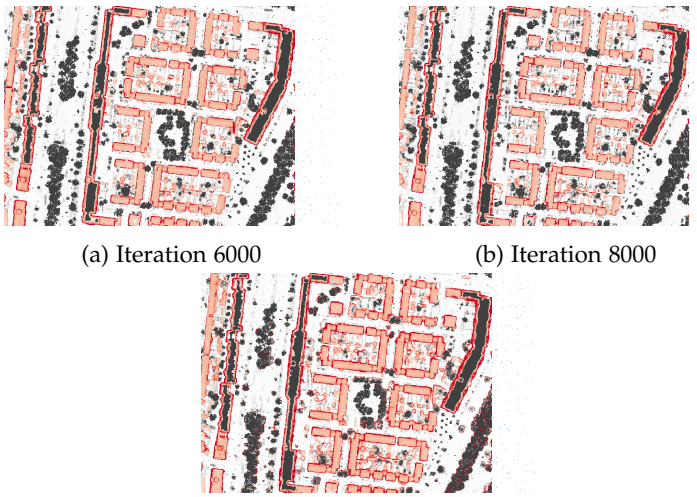
As indicated in Section 3.3, the highest cell method results in both the MedAE and the median being very high. Normally, further training would be conducted to reduce these errors. However, with the iteration onward, concave artifacts start appearing in the middle of large flat roofs in the generation results. This deformation intensifies with an increase in iteration number (Figure 4.12).

4.4.2. Threshold

Threshold is used during DSM generation as one of the constrains to convert predicted occupancy probability to occupancy status. After experiments, we find that the selection of threshold plays a crucial role in generating the edges of buildings. As shown in Figure 4.13, the residual map varies significantly at the edges for different thresholds. Notably, a threshold of 0.7 yields the best edge definition.

4.5. Limitations

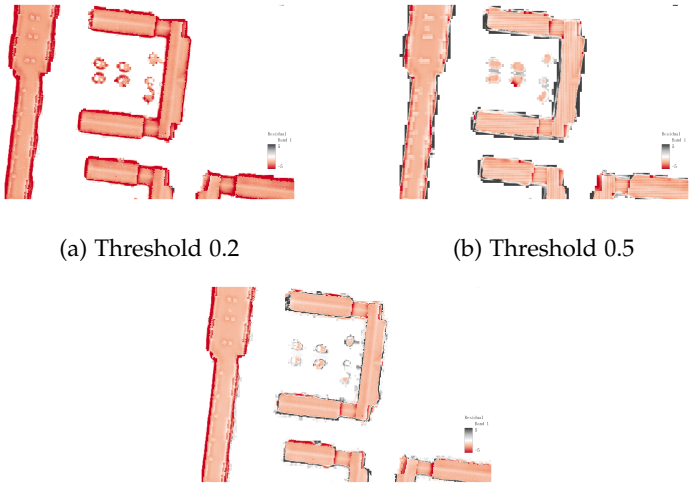
A significant limitation of implicit neural representations lies in their output, which is a continuous implicit field. Extracting specific information from this type of output is difficult because, despite its unlimited resolution, defining an efficient extraction method is challenging. In this thesis, we introduced the top-2 probability method, which yields significantly better results



(a) Iteration 6000 (b) Iteration 8000

(c) Iteration 13500

Figure 4.12.: Progressive iterations demonstrate that the artifacts on flat roofs get worse rather than diminish, and there is little increase in accuracy

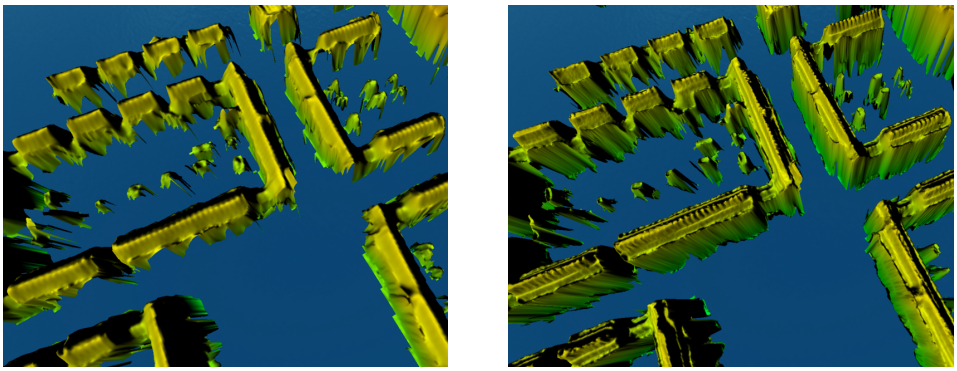


(a) Threshold 0.2 (b) Threshold 0.5

(c) Threshold 0.7

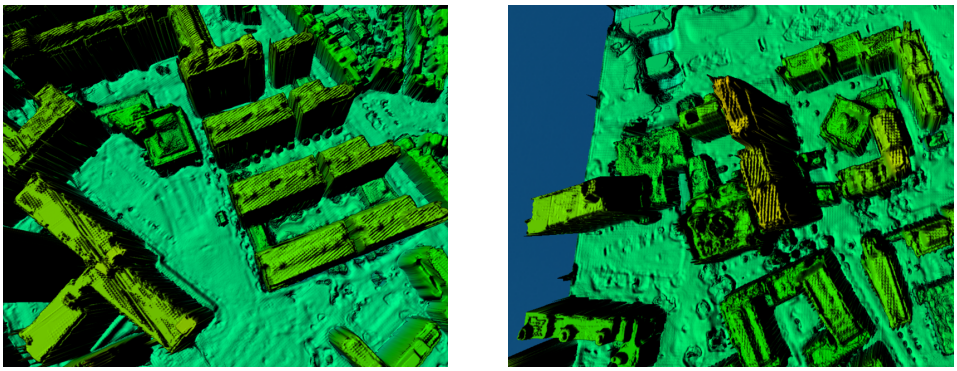
Figure 4.13.: Residual maps for different thresholds highlight the impact of threshold selection on edge clarity.

than using a simple threshold. However, it still results in a MedAE of about 0.8 m, and the generation result for roof with height change like gable roof is not ideal (Figure 4.14). Furthermore, the occupancy prediction was done cell by cell, for flat surfaces, there appears grid pattern Figure 4.15.



(a) Highest cell method (b) Top-2 probability method

Figure 4.14.: Generation result of gable roof



(a) (b)

Figure 4.15.: Grid pattern shows on the generated flat surfaces

Another issue is that the continuous nature of the field struggles with abrupt changes in height, causing buildings to appear slightly larger than their actual dimensions, creating a "buffer" area. This further reduces the accuracy of the generated DSM.

4.6. Application Scope

In our study, we demonstrate the feasibility of using implicit neural representation to generate a continuous DSM. The generated DSM can be applied to various types of analysis, such as wind simulation and flood prediction without the need for additional data processing to fill gaps in the dataset.

A complete workflow has been developed based on open-source datasets from the Netherlands. The innovative method introduced in this thesis requires only about 5000 iterations to train a model capable of producing high-quality DSM. The model can also be applied to new, unseen datasets and still maintain high accuracy. This feature can significantly reduce the need for extensive training, streamline the DSM generation for new areas, and save computational resources.

The accuracy of the generated DSM, measured in MedAE and median error, is consistently under 1 meter. The model has proven effective in filling gaps in existing DSMs, showing satisfactory results in visual assessments. Quantitative tests, where data points were deliberately removed from specific areas, demonstrate that the model achieves MAE and RMSE under 0.5 m^2, except in cases where the voids are larger than 20 m^2 or located at the edges of buildings.

The model also has extended application on vegetation classification. During the training process, terrain points are sampled from the DTM, which only includes the ground surface, excluding any vegetation or objects. The model also aims to minimize noise by masking non-essential elements, such as small-scale features like plants. However, this approach might lead to an excessive removal of vegetation details. Consequently, vegetation can be identified from the residual map where the height is lower than expected. By utilizing point cloud spectral information, detailed insights into vegetation and water content can be obtained by calculating Normalized Difference Vegetation Index (NDVI) and Normalized Difference Water Index (NDWI) for each point. Incorporating these indices in the training process improves the model's capability to accurately identify and classify vegetation, making it an effective tool for environmental monitoring and landscape management.

5. Conclusion

5.1. Research overview

To address the main research question **What are the characteristics of implicit neural representation when it's used for 3D real-scene urban area reconstruction**, we applied implicit neural representation applied to real-scene geospatial data of the Netherlands.

The sub-questions are addressed as follows:

1. What process steps are needed to adapt current geospatial data to the network?

   The open-source datasets AHN3, 3DBAG, BGT, and orthophotos are used as input data. Data preprocessing involves sampling points from a 3D city model and DTM as reference data, serving as a true occupancy reference for the network. A reference DSM is generated from AHN3 point clouds for evaluation. Masks for different land uses are obtained from BGT and classified point clouds. These masks are used during loss calculation to mitigate the varying performance of LiDAR on different land surfaces and to emphasize buildings. The DSM from AHN3 is used to remove water areas from the generated DSM.

2. What is the geometric performance of implicit neural representation when applied to real-scene urban data reconstruction?

   The best result is a merged DSM, with buildings generated using the top-2 probability method and other areas using the highest cell method. This DSM shows clear edges and corners, with most roof parts accurately generated. The residual map shows accuracy reaching 0.8 m of MedAE for buildings and 0.3 m for terrain.

3. How effective is the generalizability of the implicit representation functions on AHN3 urban data?

   The model is tested in two study areas: Eindhoven, near the training dataset, and Rotterdam, which has a different landscape and height distribution. Even towers, which are not frequently present in the training area, are well reconstructed. The evaluation shows nearly the same accuracy as in the trained area.

   The filling of void parts shows good results in both visual and quantitative checks. For numerical evaluation of the automatic void-filling ability, results indicate that for roofs with special structures, the area of missing points does not affect the generation result as long as the structure is not bisected. For flat roofs, the missing area needs to be smaller than 20 m^2 to maintain accuracy. Additionally, the model struggles to generate clear features in areas with missing points along the edges.

4. Compared to traditional methods, what are the advantages and disadvantages of using the implicit neural representation for urban scene reconstruction with open-source datasets in the Netherlands?

   Traditional DSM generation methods use interpolation algorithms or other datasets covering the same area to fill voids, often requiring frequent manual adjustments and limited data sources. In contrast, this study's method uses all open-source datasets, making it applicable across the Netherlands. Deep learning networks can learn from the geometric features of the input dataset and automatically calculate suitable parameters for DSM generation and void filling. Additionally, implicit neural representation allows for unlimited resolution and generates a continuous field, solving the problem of losing information due to discrete explicit representation. The model also shows good generalization ability, requiring no additional training when a new dataset is introduced, which greatly saves time and increases efficiency.

   However, this method requires a variety of data for training, and any differences in datasets can negatively impact the generated result, whereas traditional methods only need DSM covering the same area. The model is also prone to overfitting, with increased iterations potentially leading to artifacts on flat roofs rather than improved accuracy. The entire learning process is a black box, unlike traditional methods where all the parameters can be fully understood and adjusted as needed.

In conclusion, implicit neural representations demonstrate strong performance when used for 3D real-scene urban area reconstruction with geospatial data. They can effectively generate continuous and high-resolution DSM with high accuracy. The model exhibits notable generalization capabilities, allowing them to generate accurate data for areas with entirely different landscapes. Additionally, its ability to fill voids in the data has been validated, further supporting their efficacy in real-world applications. These characteristics underscore the potential of implicit neural representations in advancing geospatial data processing and urban area reconstruction.

5.2. Contributions

In this thesis, we developed a workflow for generating DSM using implicit neural representation with open-source datasets in the Netherlands and explored its characteristics. The contributions of this work to implicit neural representation for real scenes are as follows:

1. **Data Pre-processing Algorithms**: We developed algorithms to adapt open-source datasets in the Netherlands for use with implicit neural representation.
2. **New DSM Generation Method**: We introduced a method using occupancy probability to generate high-quality DSM at an early stage without extensive training time.
3. **Generalization Ability Exploration**: We tested the model's generalization ability on different datasets and conducted both visual and quantitative checks on the model's void-filling capability.
4. **Effect of Training Iteration Time and Threshold**: We examined the impact of training iterations and threshold settings, finding that higher iterations can cause artifacts on large flat roofs. We identified that a threshold of 0.7 is optimal for DSM generation, particularly for edge definition.

5.3. Future work

The following are potential extensions for the thesis:

1. Network input optimization

   RGB and Infrared information of the point cloud and synthetic maps can be used as input data for the neural network. The color information from the point cloud can replace orthophotos, eliminating the issues of geological alignment and feature differences caused by varying data acquisition times. Synthetic maps offer much clearer topological guidance than orthophotos, which often have interfering factors like overlapping objects and distorted grayscale histogram distribution.

2. Rotation-invariance in the representation of point cloud

   Rotation invariance can enhance the robustness of the model by seeing more data in different rotational poses and ensures the consistency in the output. In this study, rotation invariance is achieved by randomly flipping and rotating data patch before training. However, the ramdom rotation does not guarantee covering all possible orientations and there always exists unseen angle for the model. The possible solution is to generate transformation-invariant features of the point cloud by methods like local canonicalization and use it as an additional input for each point. [Yang et al., 2023]

3. Generation of other formats of 3D models

   With the continuous occupancy field established, appropriate algorithms can be employed to extract both point cloud data and 3D city models effectively. This process utilizes the continuous nature of the occupancy field to ensure complete, comprehensive, and detailed extraction, facilitating the generation of diverse 3D formats.

   In the work of Mescheder et al. [2018], they extract point clouds from the implicit field by employing a hierarchical method called Multiresolution IsoSurface Extraction (MISE). Initially, the volumetric space is discretized at a coarse resolution. Voxels with occupied status are marked active and subdivided into finer resolutions, refining the occupancy evaluations iteratively. Once the desired resolution is reached, the Marching Cubes algorithm extracts the isosurface, which is then further refined to produce a detailed and high-quality 3D point cloud. However, this methods use threshold only as constrain for extraction and can create redundant points at various height. Further work need to be done to find out how to extract desired clean surface when applied to geospatial data.

   In the work of Chen et al. [2022], they managed to combine implicit field into adaptive binary space partitioning during surface extraction from Markov Random Field (MRF), building compact and watertight building models directly from point clouds. However, this method was only applied on single building reconstruction and it's not an end-to-end network architecture. Still, this show the potential of using implicit neural representation as indicator for city model reconstruction on larger scale.

A. Web services for data retrieval

WMS and WFS commands facilitate the automatic download of orthophoto and 2D BAG data. Additionally, a specific API is utilized for downloading BGT layers, demonstrating a comprehensive approach to accessing geospatial data.

```
wget -O "path/to/output.jpg" "https://service.pdok.nl/hwh/luchtfotorgb/wms/v1_0?service=WMS&version=1.3.0&request=GetMap&layers=2019_ortho25&format=image/jpeg&crs=EPSG:28992&bbox={xmin,ymin,xmax,ymax}&width={(xmax-xmin)/resolution}&height={(ymax-ymin)/resolution}"
```

Listing A.1: WMS command for downloading orthophotos

```
wget -O "path/to/output.json" "https://service.pdok.nl/lv/bag/wfs/v2_0?service=WFS&version=2.0.0&request=GetFeature&typeName=bag:pand&srsName=urn:ogc:def:crs:EPSG::28992&bbox={xmin,ymin,xmax,ymax}&outputFormat=application/json"
```

Listing A.2: WFS command for retrieving 2D BAG data

```
# POST request to initiate the download of specific BGT layers
curl -X POST "https://api.pdok.nl/lv/bgt/download/v1_0/full/custom" \
     -H "Content-Type: application/json" \
     -d "{\"featuretypes\": [\"begroeidterreindeel\",\"waterdeel\"],\"format\": \"citygml\", \"geofilter\": \"POLYGON((xmin ymin, xmin ymax, xmax ymax, xmax ymin, xmin ymin))\"}"

# Get the status of the download request using the provided request ID
curl "https://api.pdok.nl/lv/bgt/download/v1_0/full/custom/{downloadRequestId}/status"

# Download the zip file once ready using the unique request ID
curl -o extract_plant_water.zip "https://api.pdok.nl/lv/bgt/download/v1_0/extract/{downloadRequestId}/extract.zip"
```

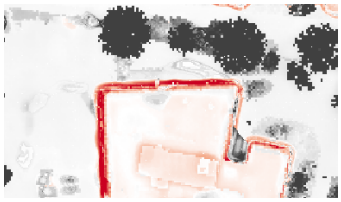
Listing A.3: API commands for downloading BGT data

B. Qualitative evaluation

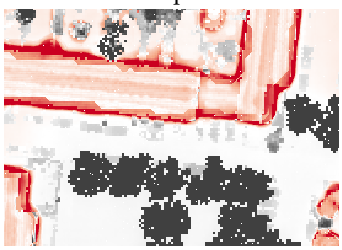
B.1. Comparisons with no photo result

To investigate the effect of including orthophotos as additional input, we also trained a model without them, utilizing only the first part of our network. However, due to the use of different models and iteration times for DSM generation, we can only conduct a qualitative comparison between results obtained with and without image guidance. Figure B.1 compares two residual maps. As observed, the building edges are significantly more blurred without the photo guidance. This demonstrates the efficiency of using photos as guidance and suggests the potential to extend the input from images to synthetic maps.

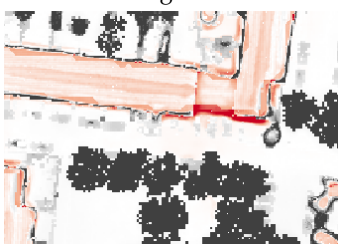
(a) Residual of Area 1 generated with no photo model



(b) Residual of Area 1 generated with regular model



(c) Residual of Area 2 generated with no photo model



(d) Residual of Area 2 generated with regular model

Figure B.1.: Building edges in different areas show better results when the model is trained with orthophoto guidance

C. Reproducibility self-assessment

C.1. Marks for each of the criteria

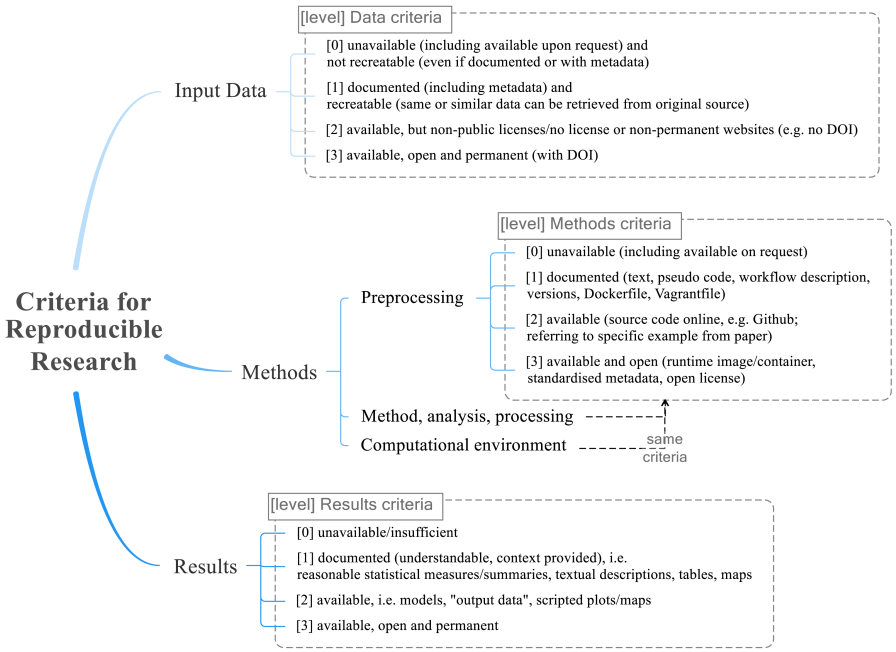


Figure C.1.: Reproducibility criteria to be assessed.

Table C.1 presents the reproducibility evaluation result of this thesis according to the criteria in Figure C.1.

Category	Criteria			Mark
	Available	Open	Permanent	
Input data	✓	✓	✓	3
Preprocessing	✓	✓	✓	3
Method, analysis, processing	✓	✓	✓	3
Computational environment	✓	✓	✓	3
Results	✓	✓	✓	3

Table C.1.: Reproducibility evaluation scores

C.2. Self-reflection

This thesis demonstrates a high level of reproducibility due to its reliance on open source datasets and transparent methodologies. The primary datasets are sourced from PDOK, a comprehensive platform in the Netherlands that provides access to various geospatial data and services.

- **Data Sources:**
  - AHN Dataset: The AHN dataset is continuously updated and provides detailed geospatial information, including point clouds, DSM, and DTM at resolutions of 5 m and 0.5 m, respectively.
  - Vector Datasets: All vector data used in this thesis are derived from the BAG and BGT datasets. These datasets offer extensive geospatial information, although they include data from different time periods, necessitating appropriate filtering.
  - Orthophotos: Each year, nationwide orthophotos of the Netherlands are taken at resolutions of 8 cm and 25 cm. These are available for free download.
- **Code Availability:** The code for data preprocessing, methodology, analysis, and processing is fully accessible on GitHub at https://github.com/StoneLee0917/DSM-with-nueral-representation-.git. This repository is open-licensed, ensuring that others can freely use, modify, and distribute the code. Data preprocessing is clearly separated from the neural network and data generation components, allowing for independent verification and replication of this step.
- **Computational Environment:** The computational environment required to run the analyses is also open source and can be executed on Google Colab. This ensures that the computational resources needed are readily accessible to anyone with internet access.

By utilizing open datasets, providing comprehensive code access, and ensuring an open computational environment, this thesis enables other researchers to reproduce and build upon the work presented.

Bibliography

Ackermann, F. (1999). Airborne laser scanning present status and future expectations. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54:64–67.

Bhattacharyya, B. (1969). Bicubic spline interpolation as a method for treatment of potential field data. *Geophysics*, 34:402–423.

Biljecki, F., Ledoux, H., and Stoter, J. (2016). An improved lod specification for 3d building models. *Computers, Environment and Urban Systems*, 59:25–37.

Boulch, A., Guerry, J., Saux, B. L., and Audebert, N. (2018). Snapnet: 3d point cloud semantic labeling with 2d deep segmentation networks. *Comput. Graph.*, 71:189–198. [CrossRef].

Chen, Z., Ledoux, H., Khademi, S., and Nan, L. (2022). Reconstructing compact building models from point clouds using deep implicit fields. *ISPRS Journal of Photogrammetry and Remote Sensing*, 194:58–73.

Chen, Z. and Zhang, H. (2019). Learning implicit fields for generative shape modeling. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5932–5941.

Choy, C. B., Xu, D., Gwak, J., Chen, K., and Savarese, S. (2016). 3D-R2N2: A unified approach for single and multi-view 3d object reconstruction. *CoRR*, abs/1604.0.

Dai, A. and Nießner, M. (2022). Neural poisson: Indicator functions for neural fields. *ArXiv*, abs/2211.14249.

Diaz, J. B. (1957). On an analogue of the euler-cauchy polygon method for the numerical solution of ux y = f(x, y, u, ux, uy). *Archive for Rational Mechanics and Analysis*, 1:357–390.

Engel, N., Belagiannis, V., and Dietmayer, K. (2020). Point transformer. *IEEE Access*, 9:134826–134840.

Geonovum (2020). Basisregistratie grootschalige topografie gegevenscatalogus bgt 1.2. Technical report, Ministerie van Binnenlandse Zaken en Koninkrijksrelaties Directoraat-Generaal Bestuur, Wonen en Ruimte Turfmarkt 147 Den Haag. Creative Commons Attribution 4.0 International Public License (CC-BY).

Häne, C., Tulsiani, S., and Malik, J. (2017). Hierarchical surface prediction for 3d object reconstruction. In *2017 International Conference on 3D Vision (3DV)*, pages 412–420. IEEE.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

He, T., Huang, H., Yi, L., Zhou, Y., Wu, C., Wang, J., and Soatto, S. (2019). Geonet: Deep geodesic networks for point cloud analysis. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6881–6890.

Huang, J. (2017). U-net implementation in pytorch. https://github.com/jaxony/unet-pytorch. Accessed: 2024-05-19.

Huang, Q., Wang, W., and Neumann, U. (2018). Recurrent slice networks for 3d segmentation of point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2626–2635. IEEE.

Kissling, W. D., Shi, Y., Koma, Z., Meijer, C., Ku, O., Nattino, F., Seijmonsbergen, A. C., and Grootes, M. W. (2023). Country-wide data of ecosystem structure from the third dutch airborne laser scanning survey. *Data in Brief*, 46:108798.

Lawin, F. J., Danelljan, M., Tosteberg, P., Bhat, G., Khan, F. S., and Felsberg, M. (2017). Deep projective 3d semantic segmentation. In *Proceedings of the International Conference on Computer Analysis of Images and Patterns*, pages 95–107, Ystad, Sweden. Springer.

Li, B., Zhang, Y., Zhao, B., and Shao, H. (2020). 3d-reconstnet: A single-view 3d-object point cloud reconstruction network. *IEEE Access*, 8:83782–83790.

Li, Y. and Baciu, G. (2021a). Hsgan: Hierarchical graph learning for point cloud generation. *IEEE Transactions on Image Processing*, 30:4540–4554.

Li, Y. and Baciu, G. (2021b). Sg-gan: Adversarial self-attention gcn for point cloud topological parts generation. *IEEE Transactions on Visualization and Computer Graphics*, 28:3499–3512.

Li, Y., Bu, R., Sun, M., Wu, W., Di, X., and Chen, B. (2018). Pointcnn: Convolution on x-transformed points. In *Advances in Neural Information Processing Systems*, volume 31, pages 820–830. NeurIPS.

Lu, Q., Chen, C., Xie, W., and Luo, Y. (2020). Pointngcnn: Deep convolutional networks on 3d point clouds with neighborhood graph filters. *Computers & Graphics*, 86:42–51.

Matheron, G. (1963). Principles of geostatistics. *Economic Geology*, 58(8):1246–1266.

Maturana, D. and Scherer, S. (2015). Voxnet: A 3d convolutional neural network for real-time object recognition. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928.

Meagher, D. (1982). Geometric modeling using octree encoding. *Computer Graphics and Image Processing*, 19(2):129–147.

Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., and Geiger, A. (2018). Occupancy networks: Learning 3d reconstruction in function space. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4455–4465.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. (2020). Nerf: Representing scenes as neural radiance fields for view synthesis.

Niemeyer, M., Mescheder, L., Oechsle, M., and Geiger, A. (2019). Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3501–3512.

Odijk, M. and Brnobic, D. (2021). Kwaliteit -en toezichtkader 2021: Basisregistratie adressen en gebouwen. Kwaliteits- en toezichtkader, Ministerie van Binnenlandse Zaken en Koninkrijksrelaties. Tot stand gekomen in samenwerking met de Werkgroep Kwaliteit BAG van het Agendaoverleg BAG.

Pan, J., Li, J., Han, X., and Jia, K. (2018). Residual meshnet: Learning to deform meshes for single-view 3d reconstruction. In *2018 International Conference on 3D Vision (3DV)*, pages 719–727.

Papasaika, H., Poli, D., and Baltsavias, E. (2009). Fusion of digital elevation models from various data sources. *2009 International Conference on Advanced Geographic Information Systems Web Services*, pages 117–122.

Park, J. J., Florence, P., Straub, J., Newcombe, R., and Lovegrove, S. (2019). Deepsdf: Learning continuous signed distance functions for shape representation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., and Geiger, A. (2020). Convolutional occupancy networks. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J.-M., editors, *Computer Vision – ECCV 2020*, volume 12348 of *Lecture Notes in Computer Science*, Cham. Springer.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017a). Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Qi, C. R., Yi, L., Su, H., and Guibas, L. J. (2017b). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, volume 30, pages 5099–5108. NIPS.

Qiu, Z., Yue, L., and Liu, X. (2019). Void filling of digital elevation models with a terrain texture learning model based on generative adversarial networks. *Remote. Sens.*, 11:2829.

Ran, Y., Zeng, J., He, S., Chen, J., Li, L., Chen, Y., Lee, G., and Ye, Q. (2022). Neurar: Neural uncertainty for autonomous 3d reconstruction with implicit neural representations. *IEEE Robotics and Automation Letters*, 8:1125–1132.

Reddy, B. R., Uttarakumari, M., S, S. S., Bency, M., D, S., Patil, K. R., and Holla, P. (2022). Machine learning based voxelnet and lunet architectures for object detection using lidar cloud points. *2022 IEEE Delhi Section Conference (DELCON)*, pages 1–6.

Saito, S., Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., and Li, H. (2019). Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2304–2314. IEEE.

Shannon, C. (1949). Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21.

Shepard, D. (1968). A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM National Conference*, pages 517–524. ACM.

Stucker, C., Ke, B., Yue, Y., Huang, S., and Armeni, I. (2022). Implicity: City modeling from satellite images with deep implicit occupancy fields. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, V-2-2022:193–201.

Su, H., Maji, S., Kalogerakis, E., and Learned-Miller, E. (2015). Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 945–953, Santiago, Chile.

Vosselman, G. and Maas, H., editors (2010). *Airborne and terrestrial laser scanning*. CRC Press (Taylor & Francis).

Wang, H., Schor, N., Hu, R., Huang, H., Cohen-Or, D., and Huang, H. (2018). Global-to-local generative model for 3d shapes. *ACM Transactions on Graphics (TOG)*, 37(6):1–10. 2018a.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., and Xiao, J. (2015). 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, Boston, MA, USA.

Xie, H., Yao, H., Zhang, S., Zhou, S., and Sun, W. (2020). Pix2vox++: Multi-scale context-aware 3d object reconstruction from single and multiple images. *International Journal of Computer Vision*, 128:2919 – 2935.

Yadav, S. (2023). Implicit vs parametric 3d shape representation. Accessed: 2023-05-04.

Yan, X., Zheng, C., Li, Z., Wang, S., and Cui, S. (2020). Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5589–5598. IEEE.

Yang, Z., Ye, Q., Stoter, J., and Nan, L. (2023). Enriching point clouds with implicit representations for 3d classification and segmentation. *Remote Sensing*, 15(1):61. * Correspondence: yeqin@tongji.edu.cn.

Yao, S., Cheng, Y., Yang, F., and Mozerov, M. G. (2024). A continuous digital elevation representation model for dem super-resolution. *ISPRS Journal of Photogrammetry and Remote Sensing*, 208:1–13.

Zhang, Y., Liu, Z., Liu, T., Peng, B., and Li, X. (2019). Reaipoint3d: An efficient generation network for 3d object reconstruction from a single image. *IEEE Access*, 7:57539–57549.

Zhao, C., Yang, J., Xiong, X., Zhu, A., CAO, Z., and Li, X. (2019). Rotation invariant point cloud classification: Where local geometry meets global topology. *ArXiv*, abs/1911.00195.

Zheng, X., Xiong, H., Yue, L., and Gong, J. (2016). An improved anudem method combining topographic correction and dem interpolation. *Geocarto International*, 31:492 – 505.

Zishu, L., Song, W., Tian, Y., Ji, S., Sung, Y., Wen, L., Zhang, T., Song, L., and Gozho, A. (2020). Vb-net: Voxel-based broad learning network for 3d object classification. *Applied Sciences*.

Colophon

This document was typeset using LaTeX, using the KOMA-Script class `scrbook`. The main font is Palatino.