# Shady Amsterdam

## Identifying the shady places and routes of Amsterdam

### GEO1101: Synthesis Project

Jessica Monahan | 4994809
Victoria Tsalapati | 5988446
Haohua Gan | 6007503
Yan Gao | 6006175
Citra Andinasari | 4994809

Delft University of Technology

# Shady Amsterdam

## Identifying the shady places and routes of Amsterdam

by



| Jessica | Victoria | Haohua | Yan | Citra |
|---|---|---|---|---|
| Monahan | Tsalapati | Gan | Gao | Andinasari |





Supervisors: Lukas Beuster & Hugo Ledoux
Date: Friday 15th November, 2024
Faculty: Faculty of Architecture and the Built Environment, Delft
GitHub: https://github.com/jsscmnhn/shady_amsterdam

Style: TU Delft Report Style, with modifications by Daan Zwaneveld

# Abstract

By providing shade for residents in urban areas, cool spaces have been shown to be essential for mitigating the effects of heat stress. In response, the Municipality of Amsterdam developed a map showing walking distances to these spaces. However, the map lacks key information on capacity, accessibility, and precise distance measurements. This project addresses these gaps by identifying quality indicators for cool places and mapping their locations and quality scores across Amsterdam. Additionally, it establishes methods for computing the shortest and shadiest pedestrian routes to these spaces, enabling efficient routing to and from any given location.

To address the research questions, the following procedures were conducted. First, shade maps of Amsterdam were created for each warm month using the Daily Shadow Pattern tool of the Urban Multi-scale Environmental Predictor (UMEP). Second, cool spaces were identified and evaluated based on accessibility, shading, usability, capacity, heat risk, and Physiological Equivalent Temperature (PET) indicators. Lastly, after obtaining and processing the pedestrian network from the Open Street Map database, shade weight was calculated for each street segment, and cool spaces were incorporated into the network, allowing users to generate datasets of the shortest and shadiest distances to cool spaces, and an algorithm that performs four different routing options: the shortest, the shadiest, and two combinations of the shortest and shadiest paths with different weighting ratios either between two locations or from a starting point to its nearest cool space.

The project produced datasets which provide insights into Amsterdam's cool spaces, their quality, and the shadiest and shortest routes to these locations. Additionally, the code to make these datasets has been made available on GitHub.

# Contents

# List of Figures

# List of Tables

# 1 Introduction

Due to climate change, urban areas have been increasingly affected by rising air temperatures. This phenomenon is known as the Urban Heat Island (UHI) effect, where microclimates in cities tend to be warmer than those in surrounding rural areas. It is caused by the physical configuration of cities, the use of construction materials, and human activities (Aleksandrowicz et al., 2020). Urban heat can impact human health and the environment, contributing to heat stress, higher mortality rates among the elderly, degraded air quality, and decreased livability of cities (Jamei et al., 2016). Studies have shown that shade plays a crucial role in mitigating the effects of urban heat, particularly during hot summer days. It enhances thermal comfort by reducing heat storage in surfaces, lowering the emission of long-wave radiation that impacts people directly (Terrani et al., 2024). Shade in urban areas is provided by the optimized design of buildings, streets, and open spaces, as well as natural shading from trees, which helps block direct solar radiation and influence surrounding air temperature and airflow (Du et al., 2020, Lee, Voogt, and Gillespie, 2018).

Public cool spaces have an essential role in delivering shade for residents in urban areas, thus providing heat stress relief, which is important for population health. To anticipate the increasing number of hot days and heat waves in the Netherlands, the city of Amsterdam developed a map that shows the walking distance to these cool spaces. However, there is no overview about the capacity, accessibility and accommodation of the existing cool spaces and the accuracy of the walking distance to these cool spaces is not always guaranteed.

To address these issues, this project has two aims. First, an overview of the locations of the cool spaces in Amsterdam and their quality score is made. The scoring criteria is based on indicators previously used by Amsterdam, as described in the CoolKit document for creating heat-resistant cities, and new indicators. Secondly, a tool is developed to calculate walking routes to cool spaces, considering both the shortest and the coolest (or shadiest) paths, and a combination of these options. This is used to create an overview of walking distances for the whole of Amsterdam, and for creating routes between specified locations.

The output of this project consists of datasets that provide more accurate distances towards cool spaces, an insight into their quality, and a GitHub repository containing code to create these datasets.

The outline of this report is as following: The scope of this project is specified in Chapter 2, which provides the problem definition. The relevant theory is reviewed in Section 3. Section 4 describes the methodology and approach used to achieve the objectives, divided into three main parts: shade maps, cool spaces, and pedestrian network. The results of these processes are shown and explained in Section 5. The overall process and results are concluded in Section 6. Finally, Section 7 provides our reflection on the method and results, and suggests recommendations for future study.

# 2

# Problem Definition

## 2.1. Project summary

The project comprises several key objectives, each contributing to the overall final outcome.

Public cool places play an important role in helping residents withstand heat stress, particularly in urban areas where they lack access to private gardens. The first objective of our research is therefore: *to identify the properties which determine the quality of cool places and use these to locate and score the cool places in Amsterdam*.

Secondly, it is important for the municipality to understand the reachability of the cool places. This information can be used in the decision-making process, for instance to determine where new cool places should be located. The second objective of our research is, therefore: *to develop a method to compute the shortest and shadiest distances for the whole of Amsterdam to cool places*.

Thirdly, for inhabitants or visitors themselves, it is interesting to know how they can reach their destination with the most shade on hot days. This would help them mitigate heat stress and UV-exposure. The third objective is, therefore: *to develop a route-making method combining the shadiest and fastest routes, from a given location to the nearest cool space or given destination.*

The final outcome shall be a GitHub repository with our code. It will include documentation about the usage of the code. Different functionalities shall be included in the program:

- **Creation of a DSM and CHM for a given area:** By using AHN pointcloud and raster data, the DSM and CHM of a user-given area can be created. This shall be used as input in the creation of shade maps. Including this functionality makes it possible to use the code for other cities than Amsterdam.
- **Creation of a shade map collection:** Using the DSM and CHM from the previous step, the shade maps for a user-chosen day and time frame can be generated. The output is a dataset of georeferenced raster files.
- **A cool places locator and scorer:** Using the shade maps and other datasets, cool places can be located and scored. The output is a polygon dataset of the cool places with their properties.
- **Dataset creation of shadiest and shortest distance to the nearest cool places for an area:** The identified cool spaces will serve as reference points for creating a walkshed, representing the distances to each cool space. This walkshed will be applied to buildings in the area, generating a dataset that includes each building's shortest and shadiest walking distance to the nearest cool place.
- **Finding the shadiest, shortest, or combined path from a given location to a given location, or the nearest cool space:** With two user inputs, starting point and either the nearest cool space or ending point, the output is the shortest, shadiest, or combination of both routes from these two locations based on a chosen time.

## 2.2. Research questions

The aforementioned objectives will be accomplished by answering the following research questions:

*'What quality indicators can be used for cool places, and based on these, where are the cool places of Amsterdam and what is their quality score?'*

This research question contains the following sub-questions:

- How can we compute the indicators for each cool place?
- What data is available for the indicators?
- What weights should the different indicators be given in the resulting quality score?

*"How can we compute the shortest and shadiest routes to cool places for pedestrians, as well as the routing to and from any given location?'*

This research question contains the following sub-questions:

- What routing algorithm should we use and how to use it efficiently with a large dataset?
- How do we ensure a correctly connected pedestrian network?
- How can we handle the different types of data representation (Nodes and edges for the paths, polygons and rasters for the cool spaces and shade)?
- How do we calculate and add shade as factor to the network?

# 3 Theory and Context

## 3.1. Overview of relevant theory

Our project shall combine the topics shade, cool places and network algorithms in one tool. To better understand these topics, we shall outline a few of their aspects relevant to our research.

To highlight the relevancy of this research, we shall explain the importance of shade in urban settings in Section 3.1.1. Here, we shall also focus on quantifying shade for usage in shade analysis. Thereafter, we shall define cool places in Section 3.1.2. Finally, in Section 3.1.3 we shall give an overview of how routing network algorithms work.

### 3.1.1. Shading

Shade provision can significantly reduce heat stress, and is therefore important for managing heat in urban climates. By obstructing direct and diffuse solar radiation, shade lowers the mean radiant temperature (MRT), which decreases the heat stress (Aleksandrowicz, 2022). The cooling effect of shade is further supported with a field study, where the thermal sensation of respondents was improved significantly in the shade (Middel et al., 2016). Moreover, shade can lower street-level temperatures and building cooling needs by reducing sunlight on surfaces like roads and buildings, decreasing heat absorption and release. Finally, utilizing shade for heat mitigation is useful because it can be manipulated by urban design. This is unlike other factors that influence heat comfort, for example air temperature. Therefore, shade provision can be deployed as a tool to help reduce heat stress (Aleksandrowicz, 2022).

To quantify shade, the Daily Shadow Pattern tool of the Urban Multi- scale Environmental Predictor (UMEP) uses a scale of 0-1 for each pixel on a raster map. A value of 0 indicates shadow for the time range the shade map was generated. A value of 1 indicates sunlight for the time the shade was generated. Values in between represent the percentage of the time range that the pixel is sunlit, for example 0.1 means the pixel was sunlit 10 % of the time (UMEP, n.d.).
Another option to quantify shade is the Shade Index (SI), which represents the proportion of blocked insolation at a given location compared to the maximum possible insolation on an unobstructed horizontal surface, measured on a scale from 0 to 1 (Aleksandrowicz et al., 2020). Insolation, or incoming solar radiation, measures the amount of solar radiation which reaches the earth surface. This index therefore also takes into account the amount of radiation, unlike the UMEP tool.

### 3.1.2. Cool Places

Cool places are areas where people can comfortably stay on a hot day, with a perceived temperature below 35°C PET during an average warm summer day, often achieved through shading. Studies show that the shading effect significantly lowers the perceived temperature and reduces heat stress for pedestrians compared to direct exposure to solar radiation (Aleksandrowicz et al., 2020). According to the CoolKit guidelines for designing a heat-resistant city, creating a balanced ecosystem that promotes significant cooling effects requires at least 200 m² of shaded space. This shading can be achieved through trees, buildings, or dedicated shading structures (HvA, 2022).

To incorporate heat resistance into spatial design, the CoolKit (HvA, 2022) outlines three main guidelines for outdoor space design. First, cool places need to be strategically located to serve a large number of people, ideally within a walking distance of 300 meters, which is especially important for the elderly. Second, there should be sufficient shade in walking areas, with at least 40% of footpaths in city centers—such as those near stations, market squares, and main shopping streets-, and 30% in residential neighborhoods. This ensures pedestrian comfort and keeps living environments accessible and attractive, particularly for the elderly, enabling them to complete their routes without overheating. Finally, the percentage of green space in each neighborhood is important, as green spaces help to moderate urban temperatures and increase evaporation capacity, reducing air temperatures and providing cooler outdoor areas during both day and night. A previous study in the Netherlands indicates that a 10% increase in green areas in a city leads to a cooling of approximately 0.5°C. (Steeneveld et al., 2011). Different neighborhood types have specific target percentages for green space.

### 3.1.3. Network

Routing network algorithms are designed to find the optimal path between two or more points in a network. The network is typically represented as a graph, consisting of nodes (points such as intersections or landmarks) and edges (connections between nodes, such as roads or paths).
There are various algorithms designed to solve the routing problem efficiently, depending on the type of network and the cost metric being optimized (e.g., distance, time, or some custom parameter like shade in the context of the project).

Dijkstra's algorithm is one of the most well-known and widely used algorithms for solving the shortest path problem in a graph. It operates by exploring all possible paths from the starting node to the destination node and selecting the path with the minimum cost (Dijkstra, 1959).

The A* algorithm is an extension of Dijkstra's algorithm, designed to improve its efficiency by incorporating heuristics. A* finds the shortest path by using a heuristic function to estimate the cost from each node to the destination. This heuristic helps guide the algorithm toward the destination, reducing the number of nodes that need to be explored (Hart, Nilsson, and Raphael, 1968).

While traditional Dijkstra's and A* algorithms focus on optimizing a single criterion (e.g., shortest distance or minimum time), real-world applications often require a balance of multiple factors. In such cases, Dijkstra's algorithm can be combined with Weighted Sum Method (WSM) which is a method to finds solution to multi criteria decision making problems. The outcome of this compound is Weighted Sum-Dijkstra's Algorithm, which can be utilized to identify the optimal path based on a combination of criteria (Hua and Abdullah, 2018).

Hua and Abdullah illustrate the methodology of the Weighted Sum-Dijkstra's Algorithm via examples. The first stage is the normalization of criteria, in which each criterion is normalized by dividing each edge's value by the total sum for that criterion, resulting in a normalized value from 0 to 1. Next, WSM values are computed for each edge by giving weights to the normalized values of each criterion and then adding them. The values of the weights determine how much each criterion will contribute to the calculation of the WSM value, which means how important each criterion will be for the determination of the optimal path. Lastly, Dijkstra's algorithm is applied taking WSM values as the weights of network edges to find the path that decreases the overall cost according to the multiple criteria.

In real-world applications of routing algorithms, high-quality geographical data is essential. The purpose is served by OpenStreetMap (OSM), which provides open, comprehensive and user-generated geographic data. OSM facilitates the construction of routing networks for diverse applications. Tools like OSMnx enable accessing and processing of OSM data,

# 4

# Methodology

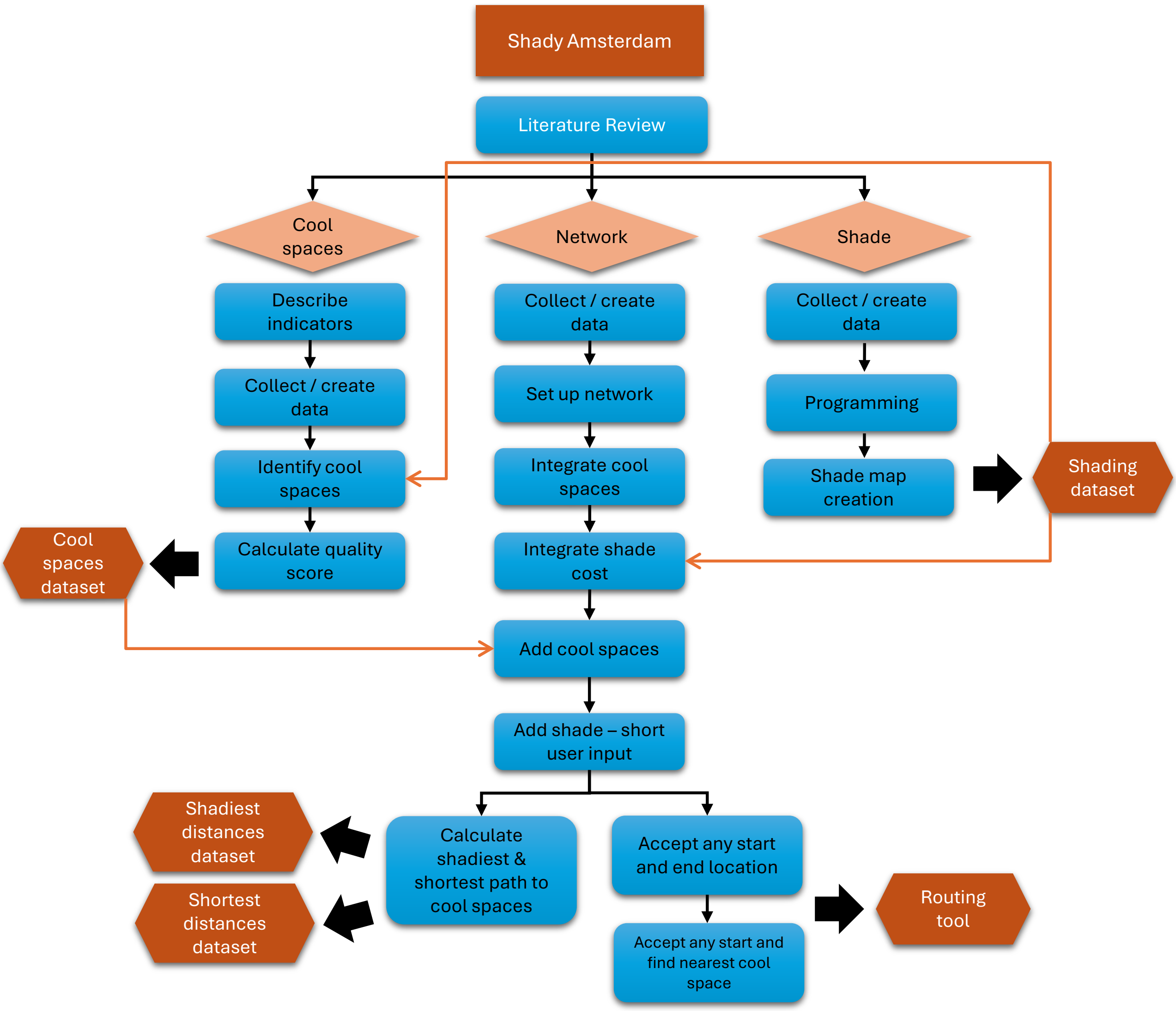



**Figure 4.1:** Overview of the research methodology with outputs.

This chapter outlines our research methodology and approach, aligned with the project's objectives and research questions. The research will be divided into three parts, allowing the literature review, dataset collection and code development to be conducted concurrently for the objectives. A visual overview is given in Figure 4.1 The first part shall cover the shade computation. For the different objectives, a database of shade data is needed. This will be elaborated further in Section 4.1. The second part shall mainly focus on the first objective, or the identification and localization of cool spaces. This shall be explained in Section 4.2. The third part covers the second and third objective, or the computation and routing of fastest and shadiest routes for pedestrians. This shall be explained in Section 4.3.

## 4.1. Shade maps

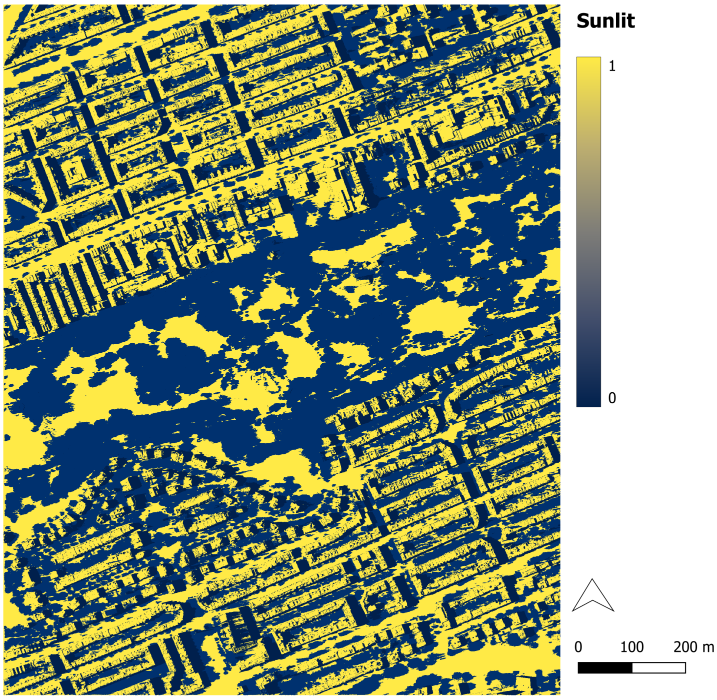



**Figure 4.2:** Example of a shade map.

To identify cool spaces and calculate shaded routes, information about the shade in Amsterdam is needed. For this, we will create a dataset of shade maps. The final shade maps will be raster files covering the whole of Amsterdam. An example of a shade map is given in Figure 4.2. They represent a pixel wise shadow analysis calculated for a specific time or time frame. A value of 0 till 1 is given, where 0 indicates the pixel is fully shadowed during the time frame and 1 fully lit. The maps will have a resolution of 0.5 x 0.5 meters and will be generated using an adaptation of the Urban Multi-scale Environmental Predictor (UMEP) Solar Radiation: Daily Shadow Pattern plugin.

Our approach is divided into two steps: collecting the required data and creating the code to process this data.

### Dataset collection

The UMEP calculation requires digital surface models (DSM) only containing the ground and building heights, and optionally a canopy height model (CHM). We shall acquire raster DTM and DSM data, and .LAZ pointcloud data from the tiles of GeoTiles (https://geotiles.citg.tudelft.nl/). The data is created from AHN data, we shall be using the most recent version on GeoTiles (AHN4). Figure 4.3 shows the division of Amsterdam in tiles and subtiles on GeoTiles. The .LAZ data shall be downloaded as subtiles and used to create the CHM by filtering out the vegetation points. The raster data shall be downloaded as tiles and used to create a ground and building DSM.

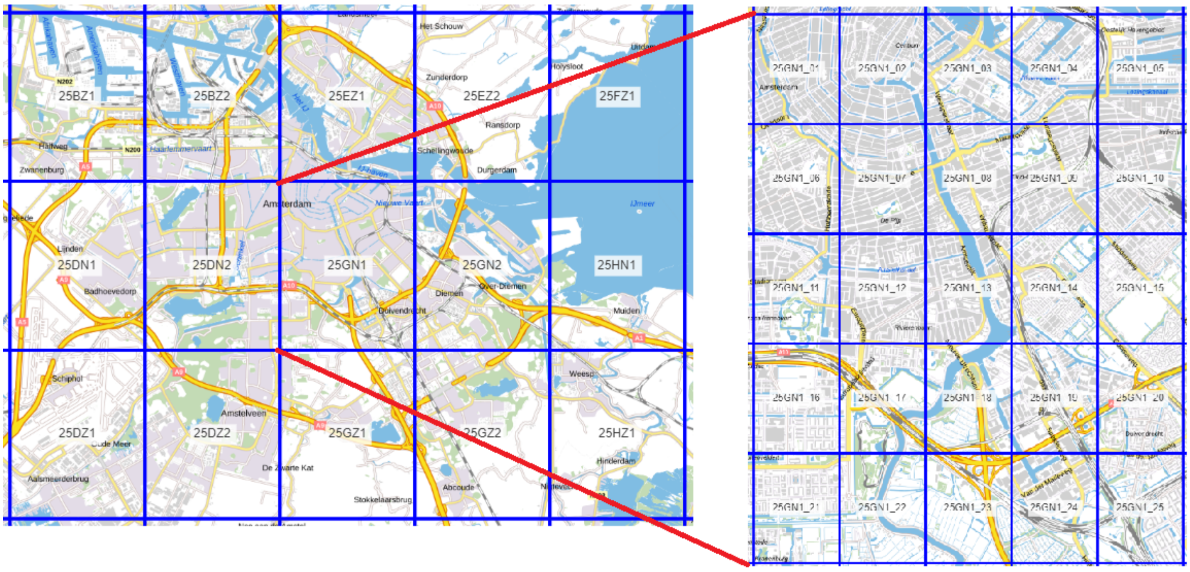



**Figure 4.3:** GeoTiles division of Amsterdam in tiles and subtiles, following the Dutch 'kaartbladen'.

Additionally, we need polygons of the buildings in Amsterdam, to retrieve the building heights from the DSM through masking. The building data shall be acquired from 3DBAG (https://docs.3dbag.nl/en/).

### Code development

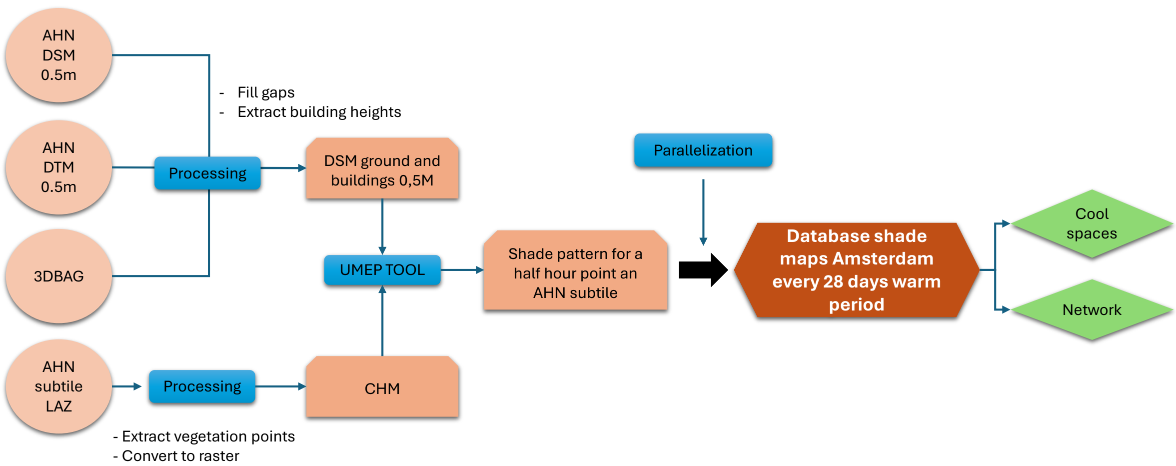



**Figure 4.4:** Overview of the shade code inputs and outputs.

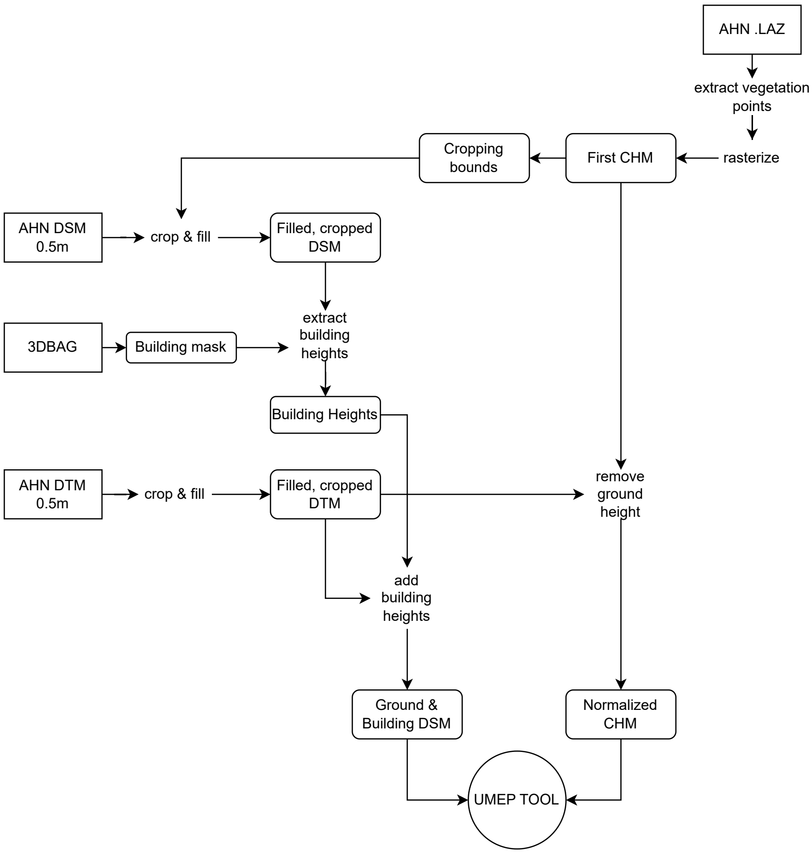



**Figure 4.5:** Overview of the shade code proceessing data for the UMEP tool.

For the code development, we shall focus on the code to calculate the shadow maps. A visual representation is shown in Figure 4.4. A more in depth visual overview of the processing before the UMEP tool is given in Figure 4.5.

First, we shall write code to create the data needed to run the UMEP *solar radiation: daily shadow pattern* tool. This code shall be able to automatically download the required .LAZ tiles, DSM and DTM tiles, and building geometries from the aforementioned sources. Vegetation points shall be extracted from the .LAZ data and rasterised. From the DSM and DTM tiles we shall create the the DSM with ground and buildings by filling the no data values, masking the building geometries, and transferring the DSM heights to the DTM usin this mask.

Secondly, for the shade calculations we shall adapt the source code of the UMEP *Solar Radiation: Daily Shadow Pattern* tool. This shall be used to generate a database of shade maps, for the hours of representative days.

A functionality that finds the shade map(s) closest to the date and time chosen by the user will be added to the Python program. The shade maps shall then be used as input for finding cool spaces and for calculating shade cost for the pedestrian network.

## 4.2. Cool space

This part aims to identify cool spaces in Amsterdam that are accessible and usable by the public. The methodology consists of three steps: First, we define cool spaces using indicators that a cool space should meet. Second, we identify potential cool spaces in the research area as candidates. Third, we score all candidates based on the indicators and filter out unqualified ones.

### Cool space definition

To define what a cool space is, we combine a set of indicators from CoolKit (HvA, 2022), shown in Table 4.1.

**Table 4.1:** Indicators of identification of cool spaces

| Indicator | Description |
|---|---|
| Publicity | A cool space should be in a public space which can be used for recreation by the citizens (e.g., greenspace). |
| Shading | A cool space should have enough shade coverage during hot daytime hours (e.g., between 12:00 and 18:00), and the shaded area should occupy an enough proportion of the total cool space area (e.g., 50%). |
| Geometrical shape | A cool space should have at least 200m² of continuous shaded area and the ratio of area to perimeter should not be less than 0.35. |
| Distance to road | A cool space should be a certain distance from a road, to avoid unusable shaded areas (e.g., greenspace between two roads) and unpleasant relaxation environment. |

To evaluate the quality of cool spaces, another set of indicators is derived, shown in Table 4.2. Further literature review is needed to determine the weights of different indicators and the threshold of the final score (i.e., cool spaces below the threshold shall be discarded). Also, the cool spaces should be evaluated in different time slots such as morning cool space scoring, afternoon cool space scoring and overall daytime cool space scoring.

**Table 4.2:** Indicators of quality evaluation of cool spaces

| Indicator | Description |
|---|---|
| Shading | The coverage of the average shaded area of a cool space during specific time slots. |
| Usability | How many rest options (e.g., benches, recreation rooms) exist in the cool space. |
| Capacity | Usage stress based on the number of dependent buildings or inhabitants. |
| Heat risk | Shows at neighborhood level the extent to which heat poses a risk to people in public spaces |
| PET | Physiological Equivalent Temperature (PET) shows human perceived temperature, ideally below 35 degrees Celsius. |

### Data collection

In order to identify and evaluate the quality of cool spaces, several datasets are collected:

Table 4.3: Data Collection for Cool Places

| No. | Dataset | Description | Source link |
|---|---|---|---|
| 1 | Shade maps | Raster data, calculated from the previous step, containing the average shading information of certain hours in representative days of the research area. | / |
| 2 | Land use | Vector data, which are polygons contain attributes of land use type. | TOP10NL |
| 3 | Road network | Vector data, which are polygons contain attributes of different road types | TOP10NL |
| 4 | Building | Vector data, which are polygons contain attributes of buildings such as function type, postcode, number of residence object | BAG PDOK |
| 5 | Public benches | Vector data, which are points representing the location of public benches | OpenStreetMap |
| 6 | Residential data | Vector data, which are polygons contain the number of total residents and vulnerable groups of resident | CBS |
| 7 | PET | Raster data, containing the PET value assess the thermal comfort of humans in various environmental conditions. | Municipality of Amsterdam |
| 8 | Heat Risk | Vector data, which are polygons contains heat risk scores | Municipality of Amsterdam |

**Cool space Process**

The process of cool spaces is divided into two steps:

- First, identify potential cool spaces based on the indicators shown in Table 4.1. Landuse map and Road network data are used for the identification.
- Second, scoring the cool spaces based on the indicators shown in Table 4.2. Then, discard the cool spaces below the score threshold. The datasets used are the Shade maps, Street furniture data, Residential data, PET and Heat Risk.

The implementation mainly used GeoPandas and Shapely Python packages to perform map calculations and produce the output cool spaces dataset. For the identification part, an intermediate result is produced and used as input for the evaluation part. The overall methodology is shown in Figure 4.6.

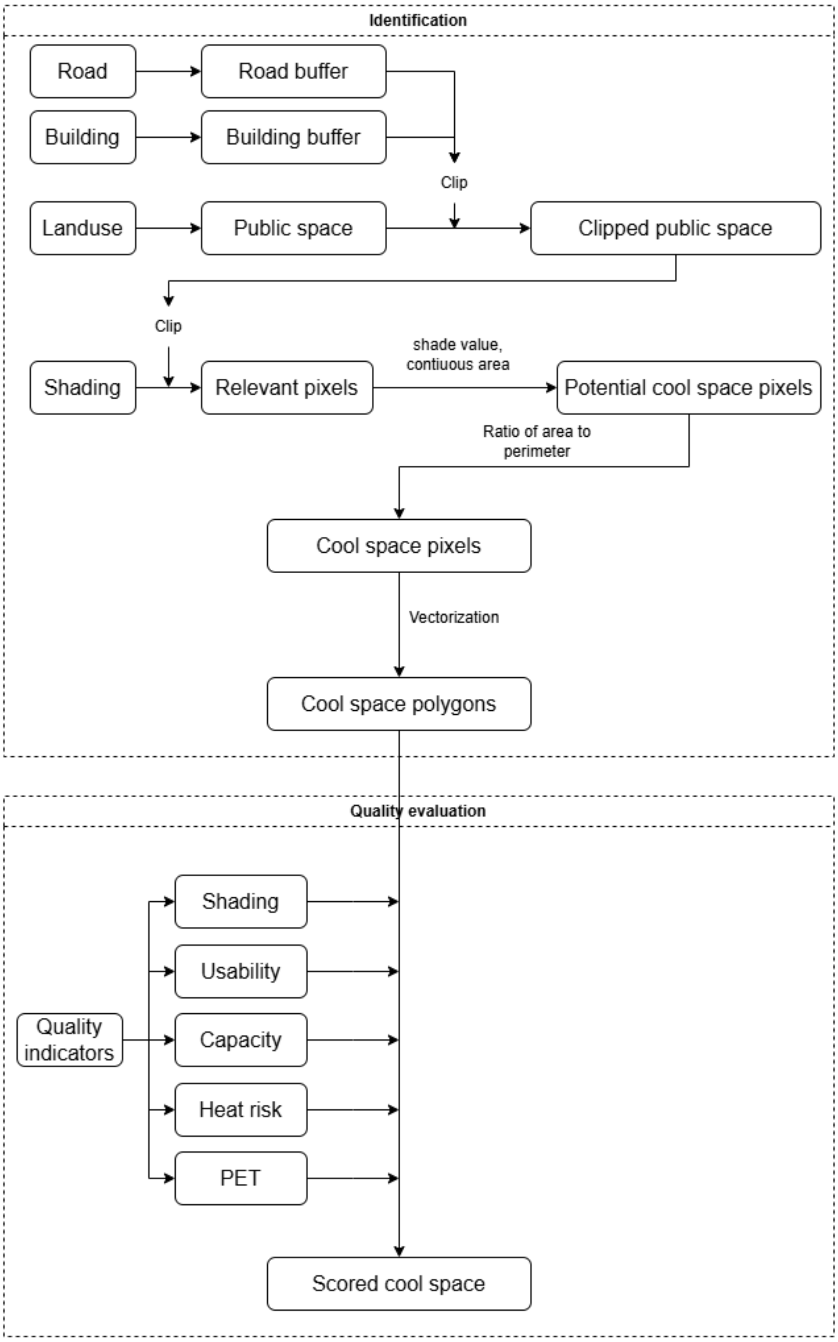



**Figure 4.6:** Overall methodology of cool space identification and evaluation

# 4.3. Pedestrian Network

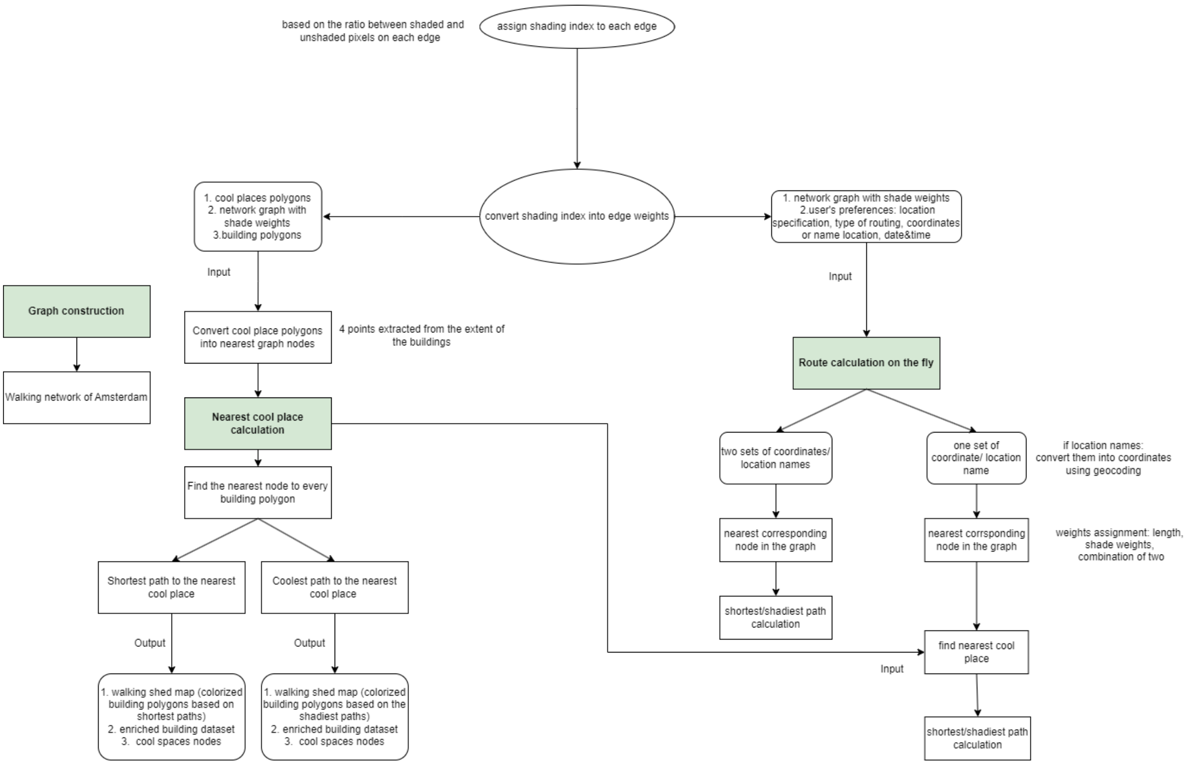



**Figure 4.7:** Overall methodology of pedestrian network

This part of the project aims to develop an executable that generates maps of walking distances to cool spaces in Amsterdam based on both the shortest and shadiest routes. It will also compute the shortest, shadiest and two balanced (combination between length and shade weight factors per edge) walking paths between two given locations or an origin and the nearest cool space at a specific time. A visual overview of the methodology to be described is shown in Figure 4.7.

### Pedestrian network and shade weights calculation

Firstly, the pedestrian network should be extracted for the study area from OpenStreetMap (OSM) using the OSMnx Python library and saved as a GraphML file, where streets are edges and intersections are nodes. More specifically, for our work, we need to acquire a network graph which includes the municipality of Amsterdam and the intermediate area between the two parts of the municipality (Figure 4.8).

**Figure 4.8:** Boundaries of the Municipality of Amsterdam

To achieve this, we first extract the network graph of the larger area named ”Metropolitan Region Amsterdam” for geocoding and convert it into two shapefiles for its edges and nodes, respectively. Next, using QGIS software, we access the boundaries of the municipality of Amsterdam through the Pdok plugin and process the data to include the intermediate area. To prevent interruptions in the network edges along the border of the polygon, we expand the boundary before clipping. After clipping, we create and utilize a Python code that performs the following tasks:

- Load an OSM walkable network graph for the Amsterdam metropolitan area.
- Read the clipped nodes from a shapefile and add them to a new, empty graph
- Identify common nodes between the OSM graph and custom node graph, creating a subgraph of shared nodes.
- Remove any isolated nodes (without edges) from the subgraph.

The resulted network graph can be viewed in Figure 4.9 and this code can be found in Appendix B.1.

**Figure 4.9:** Network graph of the study area

Subsequently, the shade weight calculation was performed aiming to quantify the amount of shade available along each street segment in the pedestrian network. For this computation, the dataset of shade maps from the first part is used as an input to determine the shading of each street segment. These raster data are aligned with the street network to ensure accurate geospatial calculations. Raster files are loaded using Rasterio.

The core of the shade weight calculation involves using zonal statistics to assess the shade coverage over each street segment. Specifically, zonal statistics computes the sum and count of raster pixel values that fall within each street segment's geometry. This process allows us to estimate the proportion of shaded pixels along each segment of the network (Equation 4.1), which determines how well-shaded a street is. A lower shade value indicates that a street segment is better shaded, making it more desirable for pedestrians on hot days.

$$\text{shade proportion} = \frac{\text{count of pixels} - \text{sum of pixel values}}{\text{count of pixels}} \tag{4.1}$$

Once the shade proportions are determined, a corresponding "shade weight" is assigned to each street segment. The shade weight is calculated to be inversely proportional to the shade proportion according to Equation 4.2. In this equation, the term $10^{-5}$ prevents division by zero in case the shade proportion is equal to zero. Edges with more shade receive a lower weight, making them more attractive during route planning that prioritizes shaded paths. For segments where shade data is missing, a high default weight is assigned, discouraging their selection in the shadiest route calculations.

$$\text{shade weight} = \frac{1}{\text{shade proportion} + 10^{-5}} \tag{4.2}$$

After calculating the shade weights, these values are added as attributes to the original pedestrian network, resulting in a shade-weighted graph. For each shade map a updated graph, which incorporates both distance and shade information, is saved in GraphML format.

### Walking distance to nearest cool spaces

Next, cool space polygons, identified in the cool spaces step, are integrated into the network graph by converting them into nodes. Each cool space polygon is represented by four boundary points rather than a single centroid to better capture the extent of the cool space area, as many of these areas have long, narrow shapes, as shown in Figure 4.10. Representing these with only a centroid would be inaccurate. More specifically, for each polygon, the nearest network node to each of its four points is identified, and any point with a distance over 50 meters between the network node is excluded. This threshold has been determined by experiments in the study area.

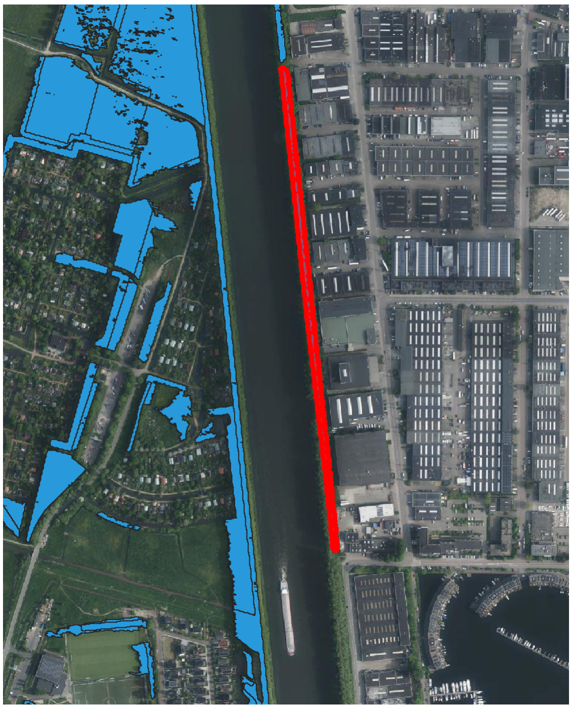


**Figure 4.10:** Example of long and narrow cool space

Once the cool place nodes are defined, the next step will be to identify the network nodes within 200, 300, 400, and 500 meters of these nodes, based on distances from the existing cool space distance map. Each edge in the pedestrian network will contain two weights: one for distance (the physical street length, used for calculating the shortest path (Equation 4.3) and one for shade (a normalized value representing the street's shading level (Equation 4.4). These attributes will enable the calculation of both the shortest and shadiest distances to the cool place nodes.

The calculation of the shade weight is explained in the documentation on the GitHub.

$$\text{normalized length} = \frac{\text{length}}{\text{total length}} \tag{4.3}$$

$$\text{normalized shade weight} = \frac{\text{shade weight}}{\text{total shade weight}} \tag{4.4}$$

```
{
    200: {node_id_1, node_id_2, node_id_5, ...},  # All nodes within 200 meters of any cool place node
    300: {node_id_3, node_id_4, node_id_6, ...},  # All nodes within 300 meters of any cool place node
    400: {node_id_7, node_id_8, node_id_9, ...},  # All nodes within 400 meters of any cool place node
    500: {node_id_10, node_id_11, node_id_12, ...}  # All nodes within 500 meters of any cool place node
}
```

**Figure 4.11:** Example of generated dictionary

The classification of the buildings based on their shortest or shadiest distance to their nearest cool space starts by generating a dictionary, such as the one shown in Figure 4.11, with distances as keys and node IDs that fall within those distance threshold from at least one of the cool place nodes as values, accounting for both shortest and shadiest paths. Next, the centroid of each building polygon will be calculated and represented, after which the nearest network node to each centroid will be identified. For each nearest building node, we will check if it is included in any cool place node's dictionary, beginning with the shortest distance category and gradually increasing it. If a match is found, meaning a building's nearest node falls within a specific distance category, the building node itself will be classified accordingly. Any points that do not meet these conditions will be assigned to the final category, for distances over 500 meters. The outputs of this process will be walking shed maps showing the shortest or shadiest distances, a building dataset with added distance attributes, and the cool space nodes dataset.

Since calculating routes for every building in the city is computationally expensive, optimization techniques are employed to ensure efficient route calculations. The optimization strategies applied are:

- KDTree for Spatial Queries: Using a KDTree spatial index accelerates spatial nearest-neighbor queries.
- Parallel Processing with ThreadPoolExecutor: This approach performs calculations for each cool place node in parallel, which is especially beneficial for computing Dijkstra path lengths across numerous nodes simultaneously.

**Routing algorithm with user input**

To implement the routing algorithm with user input, we use the OSMnx function that applies Dijkstra's algorithm. The shortest path to a cool space is calculated with edge length as the cost, while for the shadiest path, the edge shade weight is used; this way, streets with less shade are penalized, encouraging routes through cooler areas. To balance distance and shade, we implement the methodology from Hua and Abdullah, 2018 in two cases: one prioritizing the shade weight factor for the optimal path and the other prioritizing the length factor.
As described in the previous section, we first compute the normalized length and shade length for each edge following Equations 4.3 and 4.4. For the first case, we assign a weight of 0.3 to the normalized length and 0.7 to the normalized shade weight (Equation 4.5), while in the second case, we reverse these weights (Equation 4.6).

First case:

$$\text{weighted sum weight} = (0.3 \times \text{normalized length}) + (0.7 \times \text{normalized shade weight}) \tag{4.5}$$

Second case:

$$\text{weighted sum weight} = (0.7 \times \text{normalized length}) + (0.3 \times \text{normalized shade weight}) \tag{4.6}$$

The specification of the origin and destination locations, as well as the choice of routing type—whether between two locations or from the origin to the nearest cool space—is customized according to the user's preferences. The executable allows users to input either geographic coordinates (latitude and longitude) or specific location names, which are converted into coordinates via geocoding, as well the option between origin-destination or nearest-cool-space option. Once the start and end locations are set, the executable identifies the nearest nodes on the pedestrian network, ensuring that routing occurs on the street network. For the time input (date and time), the executable selects the closest date and time to the user's input by comparing it to the dates and time slots in the dataset, then loads the appropriate graph file and cool space node dataset. If the relevant field is empty, the current date and time is used.

# 5 Results

This section will cover the results of our project and their evaluation.

## 5.1. Creation of shade map database

### 5.1.1. CHM creation

From the AHN LAS data, we shall create a CHM through the following steps.

First, we have to extract the vegetation points from the LAS data. Hashim, Abd Latif, and Adnan, 2019 describe the Normalized Difference Vegetation Index (NDVI) as a method to classify urban vegetation. The NDVI is calculated as the normalized difference between Near Infrared (NIR) and Red Band (RED) values. Traditionally it has been used to calculate the density and healthiness of vegetation from satelite imagery (NASA, 2000).

Hashim, Abd Latif, and Adnan, 2019 state that the NDVI has become an important indicator for urban vegetation cover, emphasizing the necessity of high resolution images for accurate classification. The AHN LAS data meets this need, with a resolution of 10-14 points per square meter (AHN, 2020). GeoTiles provides colored point clouds, where the AHN LAS data has been combined with aerial photographs taken closest to the AHN acquisition date. This integration provides NIR and RED values for each point, though discrepancies may exist due to the merging of different data sources.

The NDVI is calculated as following:

$$\text{NDVI} = \frac{\text{NIR} - \text{RED}}{\text{NIR} + \text{RED}} \tag{5.1}$$

Based on the NDVI value, Hashim, Abd Latif, and Adnan, 2019 differentiate the following classes: non-vegetation (-1 to 0.199), shrub and grassland or low vegetation (0.2 to 0.5) and high vegetation (0.501 to 1.0). Before calculating the NDVI, we filter the points by their AHN classifications, excluding water, ground, and buildings while retaining unclassified points and those classified as low, medium, and high vegetation. Note that the AHN4 vegetation classes are empty. The NDVI threshold is set to 0, to ensure we do not miss any vegetation points. For further explanation about the chosen threshold see Appendix B.2.1. The identified vegetation points within the original point cloud are seen in Figure 5.1.

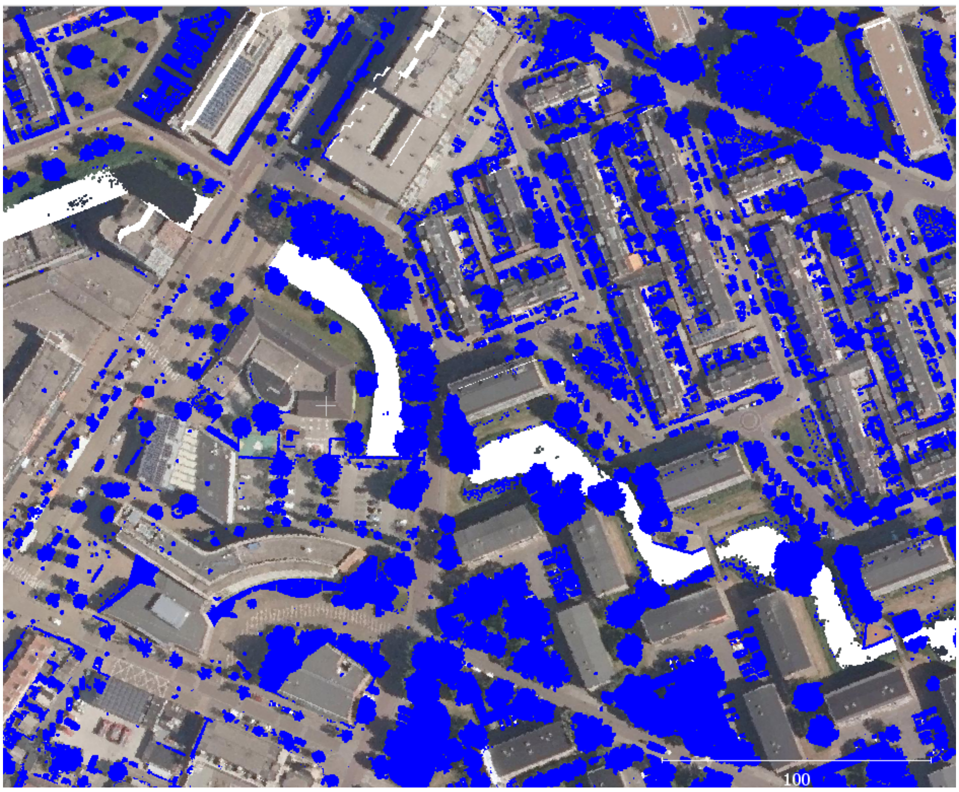
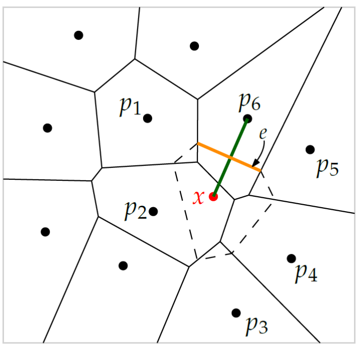



**Figure 5.1:** Points classified as vegetation (blue).

We see some issues with the identification, such shaded trees. These issues arise because the chosen method of NDVI is sensitive to the health, color and density of vegetation, which can lead to missed detections for trees. To counteract this, we set a low NDVI. This however leads to misclassifications of cars, curbs and other urban objects as vegetation. Most of these will be filtered out in the next step, where all heights below 2 meter are removed from the CHM. However, our final CHM will still contain non-vegetation points.

Finally, the extracted vegetation points must be interpolated to create a raster CHM. To achieve this, we form a Delaunay triangulation with the points. A k-d tree is employed to identify the nearest vegetation point to the center of each raster cell. If the distance is less than the specified threshold (1 meter), we perform interpolation at the center. This method is used because the Delaunay triangulation covers the whole output area. If interpolation was performed at each cell center, inaccurate vegetation heights for cells that do not contain vegetation are obtained.




$$\boxed{w_6(x) = \frac{|e|}{|xp_6|}}$$

**Figure 5.2:** The weight for the Laplace interpolant for one neighbor (Ledoux et al., 2023 p.55)

.

For the interpolation the Laplace interpolator is used. This method temporarily places a point in the Delaunay triangulation at the interpolation location. The lengths of the Delaunay and Voronoi edges of the natural neighbors of this point are used to determine the weight of the attribute (the elevation) of each neighbor. The weight of each neighbour is relative to the size of the edge of the neighbor (the orange line in Figure 5.2) and the distance between the interpolation location and neighbor point (the green line inf Figure 5.2) (Ledoux et al., 2023).

The interpolation results in the CHM map as seen in Figure 5.3

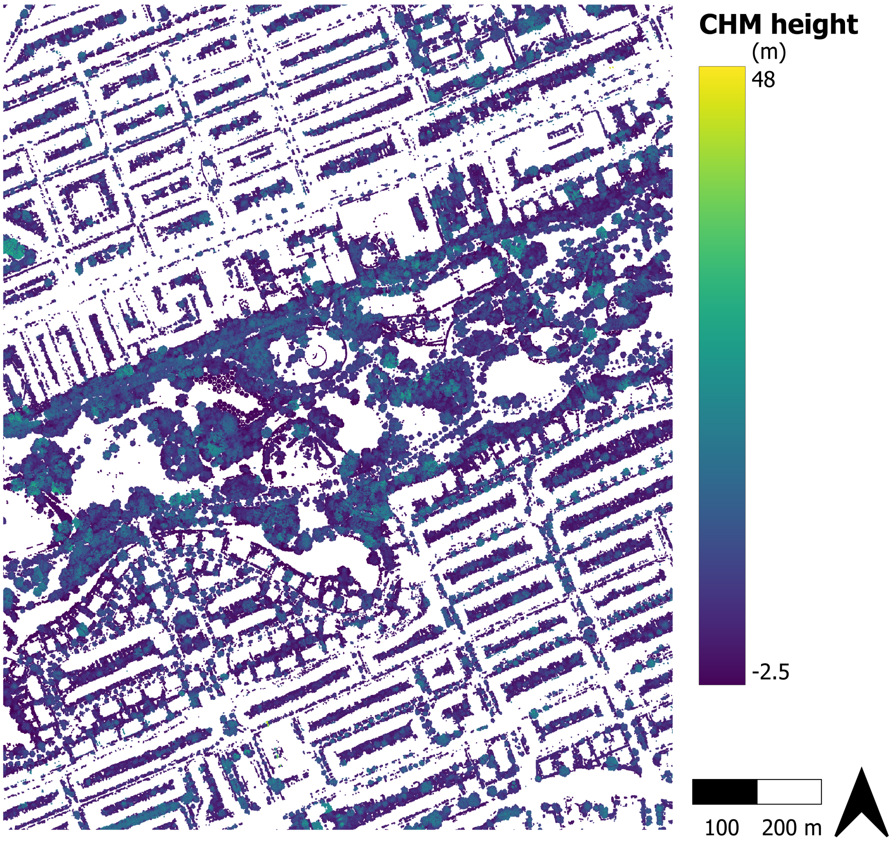



**Figure 5.3:** Output CHM of example subtile 25DN2-10.

We have also added the option to apply a median filter to the output raster, which helps to smooth out pits within the canopies. More information about this is given in Appendix B.2.2.

### 5.1.2. DTM ground and buildings creation

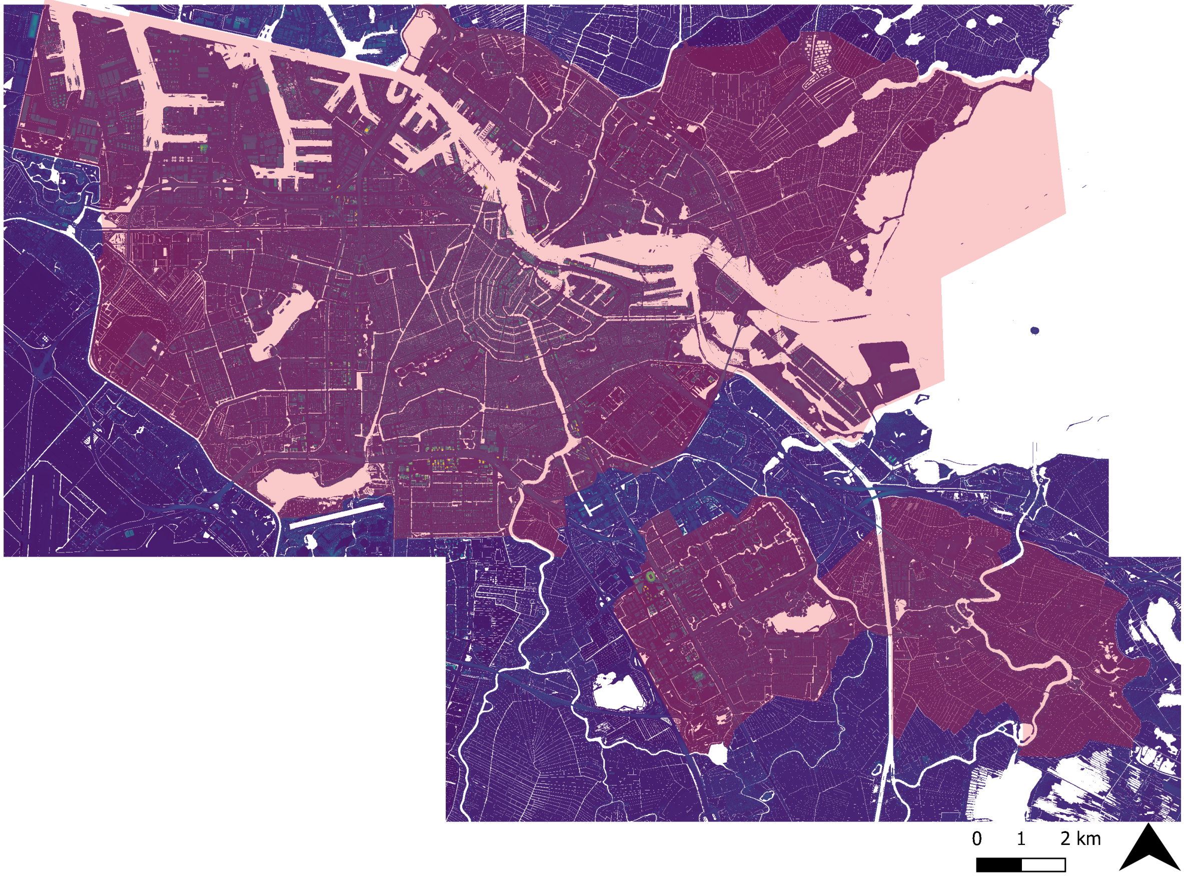



**Figure 5.4:** Merged DSM file of whole of Amsterdam.

For the creation of the DTM ground and buildings we shall use the output of the previous step, the CHM rasters, the DSM and DTM tiles from GeoTiles, and the building geometries from 3DBAG in LoD 1.3. We use this level-of-detail as it preserves the holes within building footprints, like courtyards.

The CHM rasters are initially used to crop the larger DSM and DTM GeoTiles tiles into the smaller GeoTiles subtiles. These subtiles include a buffer of 20 meters on all sides, which will be essential for the shade calculation. However, the main tiles do not include a buffer. Therefore, the main tiles are first merged into a single raster, which is then cropped for each subtile. The resulting DSM is illustrated in Figure 5.4.

Thereafter, we shall process the data per CHM subtile. First, we use the subtile to crop the merged DSM and DTM. Afterwards. we process the cropped DSM and DTM separately.

DSM

From the DSM, we need to extract only the height at buildings. To achieve this, we use the 3DBAG polygons as mask. These polygons represent the building footprints and were made based on the AHN4 data. Consequently, we aim to collect all DSM height values that fall within these polygons.

The DSM contains no-data values, sometimes in building areas. To address this, we use Laplace interpolation, the same method used for the CHM, to fill in these gaps. The cells are transformed to points with the row and column indices becoming the *y* and *x* coordinates, and the cell value becoming the z coordinate. This conversion is illustrated in Figure 5.5a and b. Interpolation for the output raster is performed only for the cells within the building mask.

To ensure that the triangulation covers the entire raster network, points must be included for each of the four corners of the array in the triangulation. As shown in Figure 5.5c, omitting a corner point leaves portions of the grid uncovered by the triangulation, resulting in undefined values during interpolation at these locations. To avoid this issue, a check is performed to confirm that corner points are populated. If any are missing, a nearest-neighbor interpolation is applied over the valid cells to fill in missing values with the nearest available ones. This approach assumes that spatial autocorrelation will provide reasonable approximations, as nearby heights are likely to be similar. However, this method can introduce slight inaccuracies due to potential local variations.

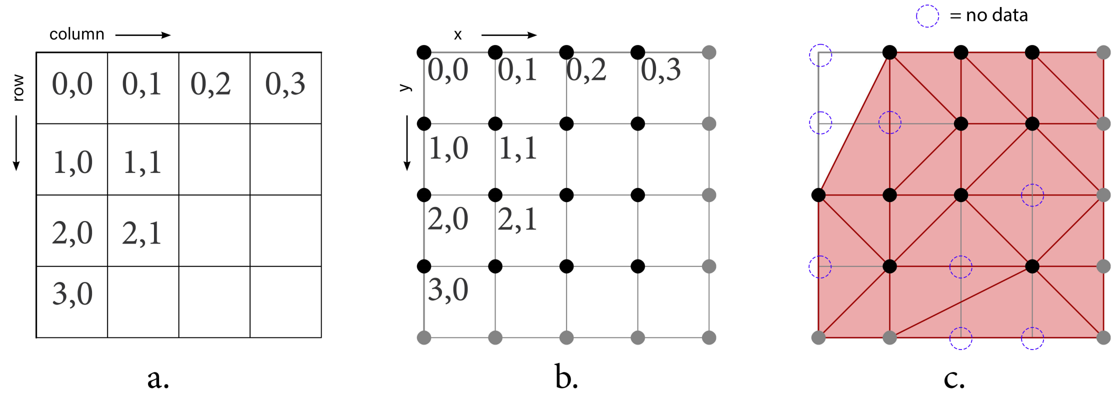



**Figure 5.5:** Converting the raster cells (a) to x,y coordinates (b). Interpolation issues can arise when the corner points are missing (c). Note: the xy-coordinates are given in yx-format.

Thereby, using row and column indices as coordinates can lead to slight mismatches in the boundaries when performing Delaunay triangulation, possibly due to floating-point errors. Since the interpolation is happening directly at the boundary for the exterior rows and columns, a minor misalignment can cause the interpolation to fail, leading to no-data values at the edges of the output raster. To address this, we remove the last column and row on each side, reducing the shared buffer area between subtitles to 19 meters.

Figure 5.6 shows the cropped DSM, the filled cropped DSM, the building footprints and the resulting raster of buildings.

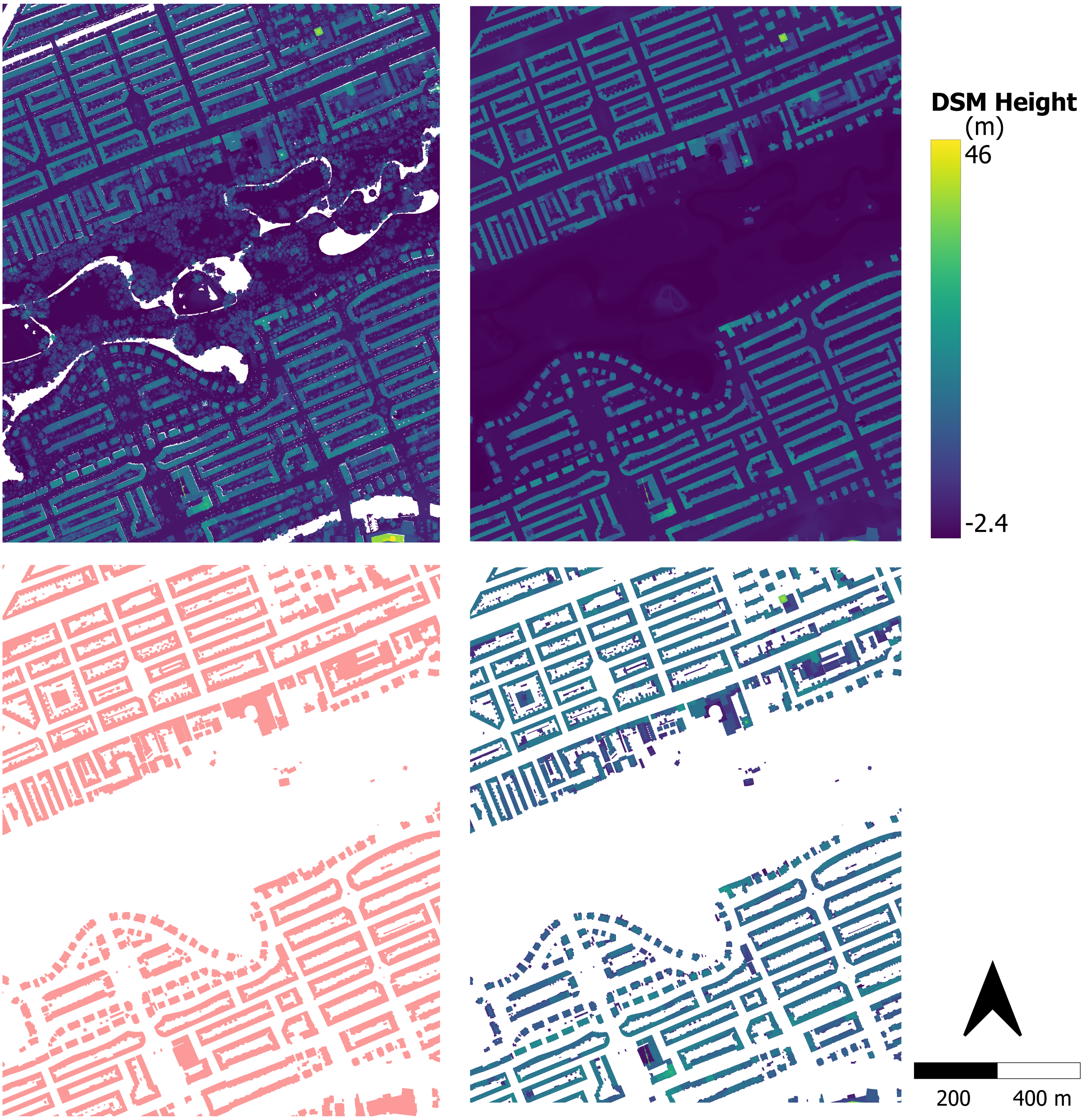



**Figure 5.6:** Cropped DSM, filled cropped DSM, building footprints and masked building heights of example subtile 25DN2-10.

One limitation of this technique is that the 3DBAG data is not as current as the Amsterdam BAG, resulting in some newer buildings being excluded. However, the AHN data, collected from 2020 to 2022, also excludes these buildings. Consequently, the generated shade maps in general will not reflect the current situation in Amsterdam. While we could use the updated Amsterdam BAG, buildings absent in the AHN data will return ground values in the mask, or trees or other objects.

DTM

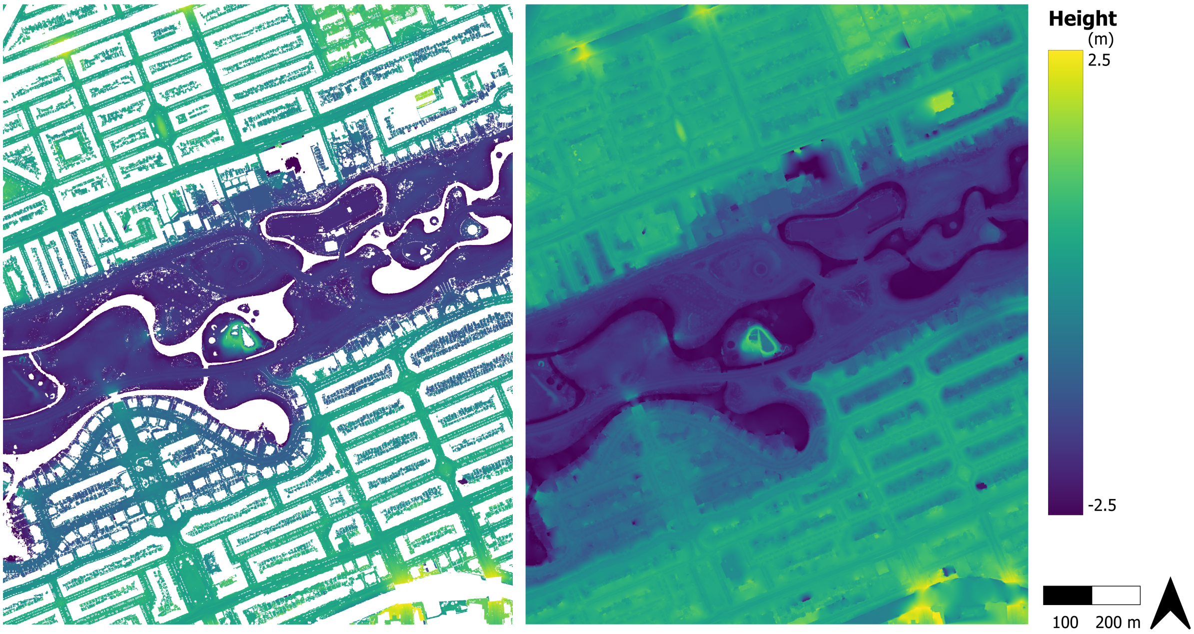



**Figure 5.7:** Before and after nodata filling DTM of example subtile 25DN2-10.

To prepare the DTM, the no-data areas have to be filled after cropping. For this, we use the same Laplace interpolation technique as described for the DSM. Figure 5.7 shows the DTM before and after the nodata filling.

In instances of large, connected no-data areas, Laplace interpolation can be slow. To address this, we have implemented an option for a faster, linear interpolation when significant no-data regions are detected in the cropped DTM. More details are provided in Appendix B.2.3. However, when running the code in parallel the other tiles can be processed simultaneously and the effect of the few slower tiles is mitigated. We recommend using Laplace interpolation for consistency and optimal results.

In this step the normalized CHM is created as well, by subtracting the filled DTM heights from the CHM. Values below 2 meters are set to 0, as vegetation of this height would not have effective shade. For example, the shade of a low bush would only cover a persons feet. Thereby, all values above 40 meter are set to 0, as the tallest tree in Amsterdam is 38.5 meters. Some CHMs may contain higher points due to the misclassification of building points to the unclassified category in the AHN LAS data. Our low NDVI-threshold then causes these points to be misclassified as vegetation. We remove the highest of these misclassfied vegetation heights. Although the misclassifications between 2 and 40 meters reduce the accuracy of our CHM, the resulting shademaps are accurate enough for our purposes as described in Appendix B.2.1.

Finally, we replace the cells in the filled DTM raster with those of the building masked DSM, at the masked locations. This results in the final CHM and the final building and ground DSM as seen in Figure 5.7.

### 5.1.3. Shade calculation

Dates and times

First, we shall determine for which days and for which hours we want to have shade data. We shall not compute all possible days to reduce computational and storage costs. We want to cover the months in which warm days can occur. According to KNMI, in meteorology a day with a temperature above 20.0 degrees is classified as a warm day (Koninklijk Nederlands Meteorologisch Instituut (KNMI), 2016). As seen in Table B.1, the previous 14 years warm days occurred between the end of March and the end of October.

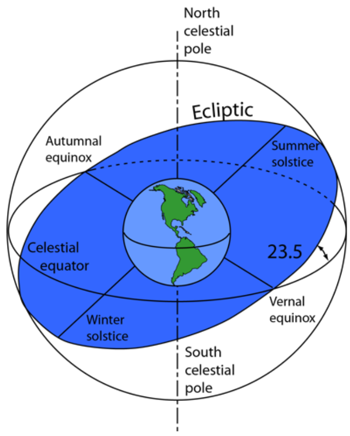



**Figure 5.8:** The ecliptic (Nave, 2004)

We shall calculate the shade patterns for four days, separated by a month (28 days). The result shall be a shade database for the dates shown in Table 5.1. The focus will be on the summer solstice of 2023, occurring on June 21st as most years, along with the days three months later. Since the sun follows a similar path before and after the solstice, as illustrated by the ecliptic in Figure 5.8, the shadow maps can be reused for the three months preceding the solstice. The approximations will not perfectly match conditions for the same half-hour. Comparisons can be found in Appendix B.2.4.

| **Month** | *March* | *April* | *May* | June | July | August | September |
|---|---|---|---|---|---|---|---|
| **Days** | *29* | *26* | *24* | 21 | 19 | 16 | 13 |

**Table 5.1:** Selected dates for the shade database. The days for March, April, May, and September are derived from shadow patterns calculated for June, July, and August, which share similar sun path characteristics.

The shade maps for March through mid-April may be less accurate due to incomplete leaf growth on deciduous trees. This limitation is acceptable since the likelihood of requiring shade during these days is low.

For the time ranges we shall collect data from 9 am till 8 pm, as the sun will be shining within this time range, and therefore shade can be desirable. The shade shall be calculated in half-hour steps.

### UMEP tool

The methodology for calculating the Daily Shadow pattern with the UMEP code is based on original work of Ratti and Richens, 1999, and the further development of this work by Lindberg and Grimmond, 2011.

The shadow casting algorithm described by Ratti and Richens, 1999, computes shadows using a Digital Elevation Model (DEM) that represents building heights. To cast a shadow, the altitude and azimuth of the Sun are specified. Shadow volumes are then calculated by sequentially translating the DEM in the azimuth direction while reducing its height based on the Sun's elevation angle. This process continues until all heights reach zero or the shadow volume shifts off the image. The final shadow map is generated by subtracting the original DEM from the shadow volume, identifying shaded and illuminated areas.

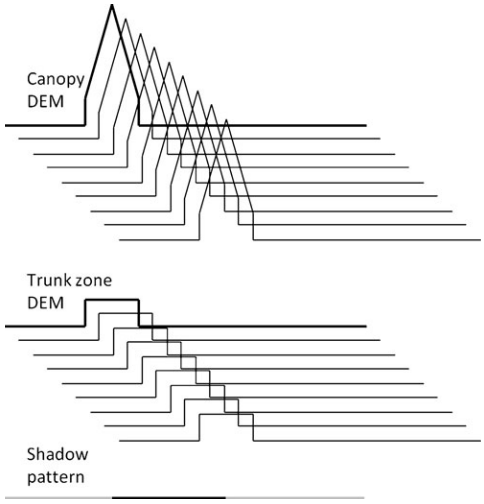



**Figure 5.9:** The repeated translation of a canopy and trunk zone DEM simultaneously reducing their heights to create shadow volumes (Lindberg and Grimmond, 2011
p.315).

To this method Lindberg and Grimmond, 2011 add the calculation of shadow volumes for vegetation DEMs, accounting for the trunk zone, the area beneath the vegetation that does not cast a shadow due to tree trunks, and the transmissivity of tree canopies. Figure 5.9 illustrates the translation and height reduction processes applied to both the canopy and trunk zone DEMs to create the shadow volumes as previously described for the building DEM.

Specifically, the UMEP code utilizes a Digital Surface Model (DSM) and, optionally, a Canopy Height Model (CHM) to calculate shadow volumes and output shadow patterns. Users input the date, time, and UTC, as well as the percentage of canopy height represented by the trunk zone. This is important because only the canopy part of the trees casts shade; the trunk below does not contribute. Users also specify the transmissivity of tree canopies, as trees allow some light to pass through.

The code computes the sun's precise position at the specified time(s) for the location of the input data. Using the previously described methodology, it calculates the shadow pattern for each given time. Additionally, we have implemented an option for users to define a start and end time for the calculation of multiple shade maps within a single day, allowing for more flexible calculations.

#### Shade Map

For one subtile, we can now create a shade map using the adapted UMEP tool code, the CHM and the DSM with buildings and ground. Figure 5.10 shows the shade map for a subtile at 13.00.

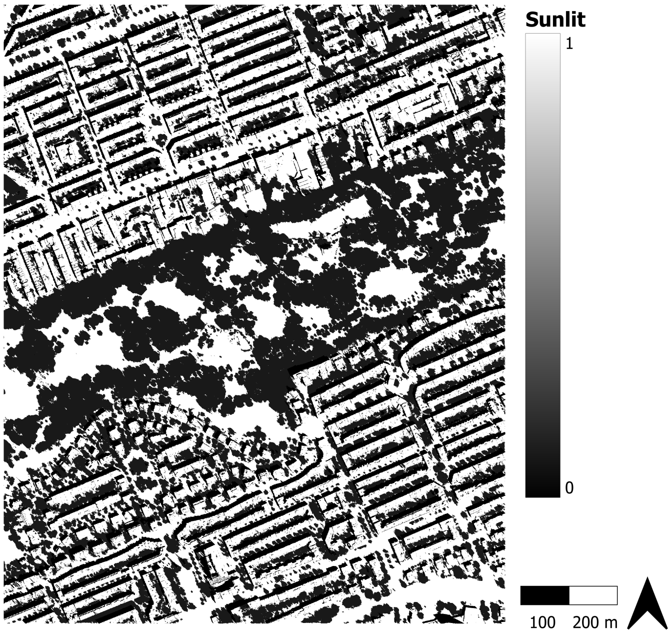



**Figure 5.10:** Shade map of example subtile 25DN2-10.

For the trunk height, we use the default value of 25% from the UMEP tool due to the lack of data on the average trunk height of all vegetation in Amsterdam, and no means to create a trunk height DSM. For transmissivity of light through vegetation, we set the value to 10%. The UMEP tool's default is 3%, based on research indicating that the average transmissivity of direct solar radiation through foliated and defoliated tree crowns ranges from 1.3% to 5.3% and from 40.2% to 51.9%, respectively (Konarska et al., 2013). To address potential inaccuracies in our CHM, we opted for a higher transmissivity value.

### 5.1.4. Shade database creation

The previous sections described all the steps needed to create the shade maps for one subtile for one day. However, we need to calculate the shade for multiple days for 233 subtiles. Therefore, the code needs to be adapted to calculate and save data efficiently in a way that is easy to query. Thereby, the separate shade maps for each subtile should be merged into one shade map for the whole of amsterdam

#### Merging shademaps

Merging the shade maps is a straightforward process. First, all subtile maps covering the same day are collected by examining their time suffixes. Next, the bounds for the output merged shade map are determined by finding the minimum and maximum values from all the subtile maps. An array of no-data values is then created to match this shape. The subtile maps are placed one by one into the array. The output array values are replaced with those from the subtile if the current array value is marked as no-data or if it is higher than the incoming value. This approach makes use of the buffer surrounding the subtiles, where one subtile may contain shade information from buildings that an overlapping tile lacks. By always replacing values with the lowest (i.e., shadiest) output, we ensure that the final map accurately reflects all shade. An example of a merged shade map is shown in Figure 5.11.


**Figure 5.11:** Shade map of Amsterdam

#### Database structure

For each intermediate step in the shade map creation, all output subtile files are saved in a directory automatically named after the main tile. The GeoTiles main tiles are always named in the pattern 00XX0, for example "25DN1". The subtiles are named in the format {main tile}_{subtile number}, with each main tile containing 25 subtiles. By organizing all outputs, traverse by main tile.

For the final output, the name of the merged shademaps contain information about the date and time they are created for, IN THE FORMAT {user chosen name}_{YYYMMDD}_{time},. The shade maps of one day are placed in the same directory.

#### Parallelization

The tile-based calculations allow for easy parallelization by processing multiple tiles simultaneously. In the creation of the initial CHM, DSM, and normalized CHM, as well as during shade calculation, a list of subtile files is fed to a *ProcessPoolExecutor*. Each subtile is processed independently across a specified number of worker processes, reducing computation time.

## 5.2. Cool Space

### 5.2.1. Identification

#### Creation of the public space data

The public space data is created from three datasets: land use data, road data and building data.

First, we use the "typelandge" (short for "typelandgebruik", translation "type of land use") attribute of the land use data to filter out irrelevant land use types including transportation areas and graveyard areas.

Second, distance buffers for road data and building data are created. For road data, the buffer is created according to different road types shown in Table 5.2, which are from the attribute "typeweg". For building data, the distance buffer is set to four meters. It is important to emphasise that for the road data in this case, the distances for road polygons are created based on a road type attribute. However, we expect the user input to have a buffer attribute specifying the distance buffer number for each polygon.

Third, two buffered data are used to perform clip operations on the filtered land use data to get the initial public space data.

Fourth, a negative two-meter buffer is set to filter out areas which are thinner than four meters. This step aims to filter out thin areas which are due to the two clip operations.

Finally, the public space data is filtered by two criteria: 1. The area of the public space is not lower than 200$m^2$; 2. The area ratio of area to perimeter is not lower than 0.35, to get the final public space data, shown in Figure 5.12.

**Table 5.2:** Buffer sizes for different road types

| Road type (Dutch) | Road type (English) | Buffer size (m) |
|---|---|---|
| autosnelweg | highway | 20 |
| hoofdweg | main road | 15 |
| regionale weg | regional road | 10 |
| lokale weg | local road | 10 |
| straat | street | 5 |
| parkeerplaats | parking lot | 5 |
| overig | others | 0 |

### Shaded area creation and evaluation

The goal of this step is to identify how many shaded areas a public space contains within a specific time range. Four different time ranges are set:

- Daytime: from 9:00 to 18:00
- Morning: from 9:00 to 12:00
- Afternoon: from 12:00 to 16:00
- Late afternoon: from 16:00 to 18:00

First, select corresponding shade maps according to the time range. For example, given the fact that the shade maps start from 9:00 and have a time interval of 30 minutes, for daytime, corresponding shade maps indices are from 0 to 17, 18 shade maps in total.

Second, for each shade map, public space data is used to perform a clip operation and select continuous shaded areas which have pixel values lower than 0.5 and also within the public spaces. The continuous shaded areas also need to meet the same criteria in public space data creation.

Third, each public space now has shaded areas for all the shade maps within the time range. Then, two indicators are evaluated: shade coverage in terms of time (A), and shade coverage in terms of area ratio (B). For A, the value is calculated as:

$$A = \frac{N}{M} \tag{5.2}$$

Where $N$ represents the number of shade maps that produce valid shaded areas, and $M$ is the total number of shade maps within the time range. For example, for daytime range, $M = 18$. Assume a public space has $N = 9$, then $A = 0.5$, which means it has valid shaded areas during $50\%$ of the time range.

For B, the value is calculated as:

$$B = \frac{Area_{avg}}{Area_{tol}} \tag{5.3}$$

Where the $Area_{avg}$ represents the average shaded area within a public space, and the $Area_{tol}$ is the area of the public space. For example, if a public space has an area of $500m^2$ and it has an average shaded area during the daytime range of $300m^2$, then $B = 0.6$, which means an average of $60\%$ of the area of this public space is covered by shade during the daytime hours.

Both indicators will be classified into 4 classes:

- 0: $coverage < 50\%$
- 1: $50\% \leq coverage < 70\%$
- 2: $70\% \leq coverage < 90\%$
- 3: $90\% \leq coverage$

Corresponding results are shown in Figure 5.13 and Figure 5.14.

### Output of identification process

The shaded areas and indicators calculated in previous steps are used as inputs for the evaluation process. Therefore, an output vector data containing these attributes is created.

If a public space has at least one valid shaded area during the specified time range, it will be output as part of the result, otherwise, it will be discarded.

There are two geometry types that can be chosen: land use geometry and public space geometry. If land use geometry is chosen, the output vector data will use initial land use polygons as geometry, and the public space geometry is stored as a Well Known Text (WKT) attribute, and vice versa. Furthermore, all shade areas will also be stored as WKT attributes in the output file, as shown in figure **??**.

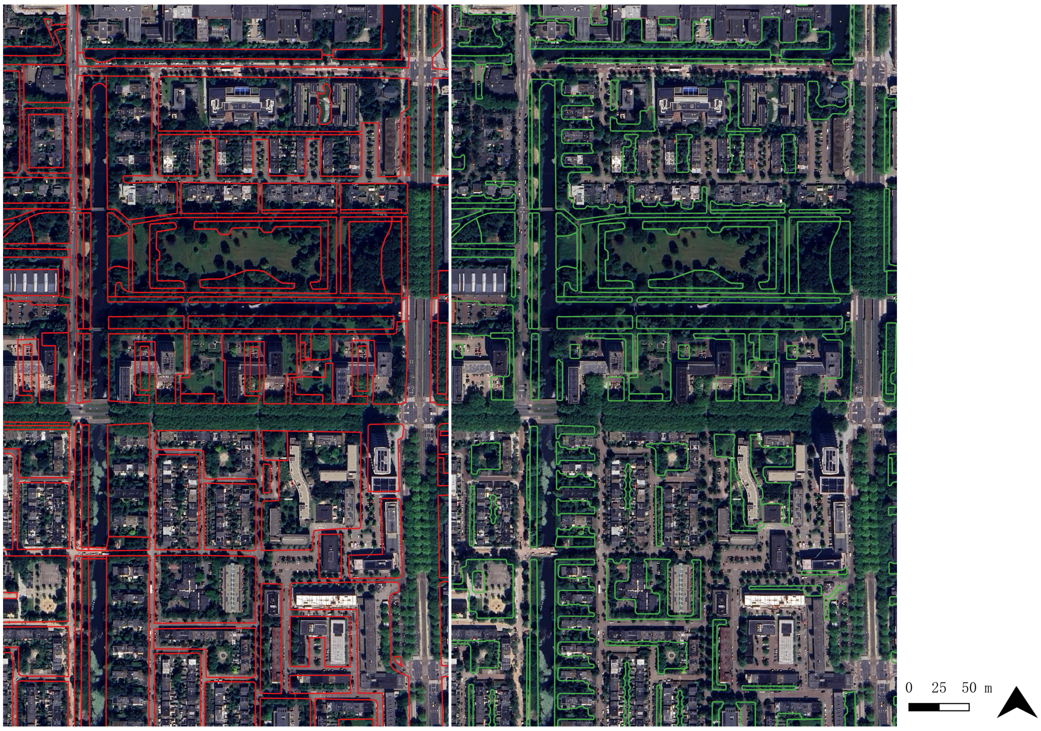



**Figure 5.12:** Comparison of land use data and public space data. Left: land use data. Right: public space data


**Figure 5.13:** Evalutaion result of shade coverage indicator (area ratio), daytime hours

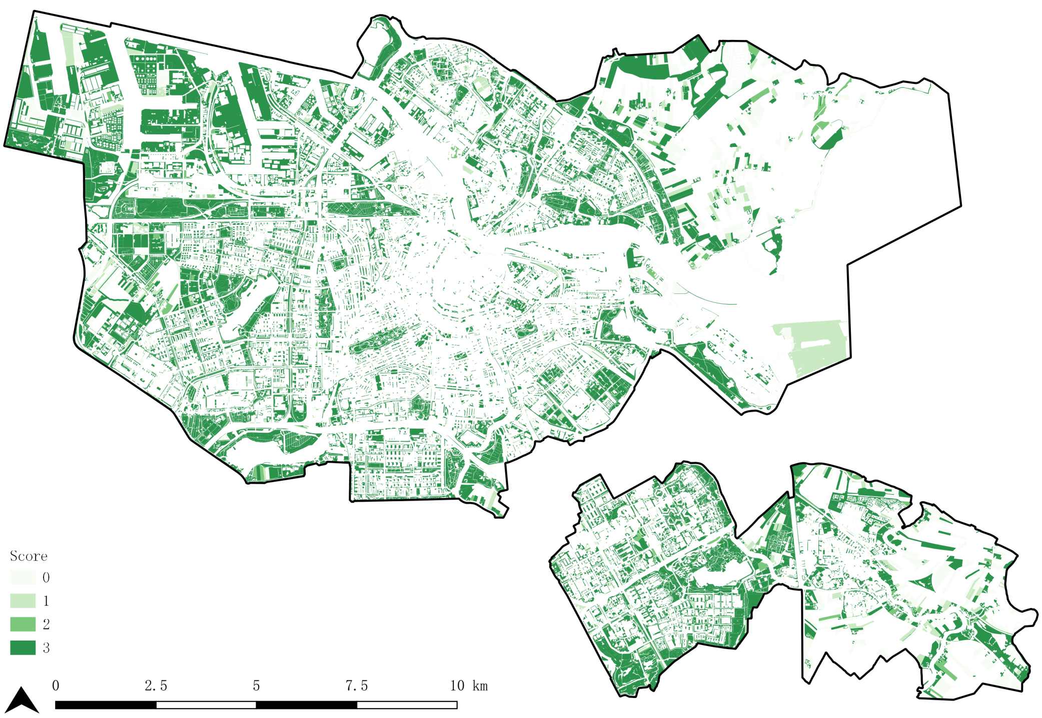



**Figure 5.14:** Evaluation result of shade coverage indicator (time), daytime hours

| Month | Cool Spaces |
|---|---|
| June | 15114 |
| July | 14065 |
| August | 14613 |
| September | 15507 |

**Table 5.3:** Identified cool spaces per month

### 5.2.2. Evaluation

After getting the polygon of cool spaces candidates that contain 18 different shade geometries columns in WKT format, which refers to every 30-minute interval between 9.00-18.00, we iterated each shade geometry within the cool space candidates to assess them based on the indicator. The resulting scores were then aggregated and averaged for each cool space candidate. Based on these averages, we determined whether each candidate qualifies as a recommended cool space. The detailed steps are outlined below.

#### Shade Evaluation

The level of shading significantly enhances thermal comfort for visitors (Mumcu and Yılmaz, 2016). There are two shade indicators, as already explained in the previous process 5.2.1, shade coverage based on time, and shade coverage based on area ratio. Both of those indicators will be assessed to create three different recommendations: first, it scores both of them equally, and the second and third recommendations will score them individually.

#### Capacity Evaluation

To determine the capacity of cool spaces, several steps need to be followed. First, the residential dataset and the building dataset are combined based on postal codes. For residences sharing the same postcode, the total population is divided by the number of residences to calculate the average number of residents per residence. The total number of residents for each building is then obtained by summing the residents from all residences within the building.

It is important to note that the CBS data may lack accuracy, as population information may not be available for certain postal code areas, resulting in some buildings having no recorded population. This may also occur if the buildings are offices rather than residential structures.

Once the number of residents per building is established, the nearest cool places within a walking-distance buffer are identified for each building. Buildings that share the same cool place ID are grouped together and merged with the cool space datasets to calculate the resident capacity of each cool place. Assigning a cool place ID is essential to avoid redundancy or duplication in resident calculations. According to the guideline that every person requires at least 10 $m^2$ of green space (Laan and Piersma, 2021), the capacity evaluation is calculated by dividing the area of the cool place by 10 and then subtracting the resident capacity of the cool place from this value. A negative result indicates that the cool place exceeds its service capacity, while a positive result suggests that there are available spots for additional visitors beyond its service capacity. However, it is important to note that the capacity values may not be entirely accurate, as it is common for the geometry of a cool space to be surrounded by that of others. In such cases, the surrounding cool space does not get any residents assigned, as the other cool spaces will be closer.

$$Capacitystatus = \frac{Area}{10} - Resident \tag{5.4}$$

| | June | July | August | September |
|---|---|---|---|---|
| **Under Capacity** | 12427 | 11414 | 11975 | 12960 |
| **Over Capacity** | 2687 | 2651 | 2638 | 2547 |

**Table 5.4:** Capacity Evaluation in June-September

### Public bench Evaluation

The public bench has an essential role in attracting people to visit an urban space and encouraging them to stay longer. Availability of seating can provide comfort in public spaces and is an important indicator of whether the public space is hospitable or not. Various studies show that an increasing number of seats significantly increases the number of visitors (Mumcu and Yılmaz, 2016). To assess the public benches in cool places, we count the number of benches from the dataset present within each cool space polygon to determine the total seating available. However, due to limitations in the data from OSM, many cool spaces are missing seating information, with almost half lacking any recorded seats, as illustrated in 5.5 below. Interestingly, there are 32 cool spaces that have more than 60 seats.

| Number of seats | Cool Spaces |
|---|---|
| 0 | 7837 |
| 1-20 | 6301 |
| 21-40 | 334 |
| 41-60 | 105 |
| >60 | 32 |

**Table 5.5:** Number of seats in cool spaces

### Heat Risk Evaluation

Heat risk evaluations are derived from the Heat Risk map of Gemeente Amsterdam, showing the relative risk heat poses to people in public areas on a neighborhood level. It integrates perceived temperature data with three key factors—exposure, sensitivity, and adaptability—to assess heat risk in various neighborhoods (Gemeente Amsterdam, 2024). These factors are defined as follows:

1. Exposure refers to the degree to which residents are subjected to heat, taking into account factors such as the percentage of shaded areas along walkways.
2. Sensitivity shows the potential for people to suffer from heat effects, such as by considering the proportion of residents over 65, who may be more vulnerable to high temperatures.
3. Adaptability assesses how well residents might cope with heat stress in their area, measured by factors like the walking distance to nearby cooling spaces, such as parks.

The heat risk scores are classified into five categories, with 1 being the lowest, indicating the least vulnerability to heat risk, and 5 being the highest.

### PET (Perceived Effective Temperature) Evaluation

To ensure the comfort of cool space, the PET map derived from Gemeente Amsterdam is also added to our evaluation. This map illustrates the perceived temperature at the living level, serving as an important indicator of heat's impact on human health. It is created by considering various factors affecting perceived temperature, including solar radiation, wind, and humidity (Gemeente Amsterdam, 2024). The PET is classified into three parts for scoring based on thermal perceptions ranges (Lin, Matzarakis, and Hwang, 2010), PET below 30 (slightly warm) is scored as 1, between 30-35 (warm) is scored as 0.8 and for more than 35 (hot) is 0 as it is not recommended as cool space( HvA, 2022). However, since the PET dataset was created based on temperature on July 1st, thus PET criteria are only used for June and July, which is not suitable for the other months.

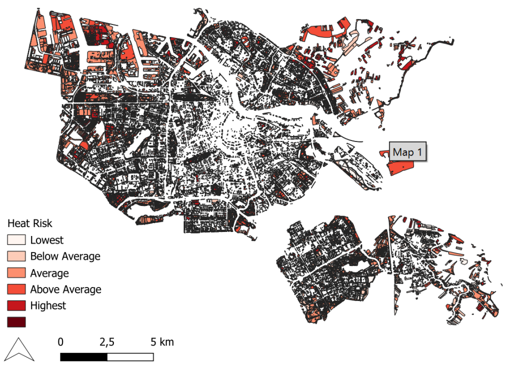



**Figure 5.15:** Heat Risk Level


**Figure 5.16:** PET

## Recommendation

After evaluating all the cool space indicators, we weigh their scores based on their priority to make a recommendation of cool space as shown in table 5.6. Most indicators have equal weight except for Shade with the highest weight as it is the most essential indicator for cool space. Since PET is only used for June and July, for August and September the weight of PET is transferred to Heat Risk due to a similar indicator. Before weighting their scores, their values are normalized, so their scores can be calculated together. PET, Heat risk and shade, however, are adapted individually according to their classification. After summing all the indicators along their weight, the recommendations are divided into three parts: *Highly Recommended* if the final score is higher than 0.6, *Recommended* for a score between 0.3-0.6 and *Not recommended* if the final score is below 0.3 Additionally, extra condition is added for *Not Recommended* if the shade coverage is 0 (has less than 50% coverage). The results of all recommendation maps from June to September are shown in the Appendix B.4. Our results, as shown in Table 5.7, depict that September has the highest number of highly recommended cool spaces, while June has the highest number of not recommended cool spaces.

| Indicator | Weight |
|---|---|
| Capacity | 0.15 |
| Public bench | 0.15 |
| Heat Risk | 0.15 |
| PET | 0.15 |
| Shade | 0.4 |

**Table 5.6:** Weight Indicator

| | June | | July | | August | | September | |
|---|---|---|---|---|---|---|---|---|
| **Highly Recommended** | 7395 | 48.93% | 7518 | 53.45% | 8365 | 57.24% | 9431 | 60.82% |
| **Recommended** | 4267 | 28.23% | 4260 | 30.29% | 3907 | 26.74% | 3569 | 23.02% |
| **Not Recommended** | 3452 | 22.84% | 2287 | 16.26% | 2341 | 16.02% | 2507 | 16.17% |

**Table 5.7:** Cool Space Recommendation during June-September with Percentage *note: only June and July using the PET evaluation

## 5.3. Pedestrian Network

### 5.3.1. Walking shed Maps

The generated walking shed maps show the classified according to the shortest and the shadiest distance to their nearest cool place building polygons of the study area. The walking shade maps depicting the datasets of the shortest and the shadiest distances on 21st June and at 2 pm for the overview of the study area are presented in Figure 5.17 and 5.18. As a first observation, we can notice that in both of the maps the areas with buildings of the longest shadiest category that stand out are the one around the central train station and the industrial area to northeast. This is because cool places nodes there are significantly less compared to other areas. For the case of the industrial area, the reason for this result is that the walkable network there is not so extensive as in other places in Amsterdam. On the other hand, the fact that in the area around the central station of Amsterdam the network is more dense indicates that there is a serious lacking shades and cool spaces, and as a result a person starting their route from a building in this area to the nearest cool place should potentially walk a long distance either they pick the shortest or the coolest route.

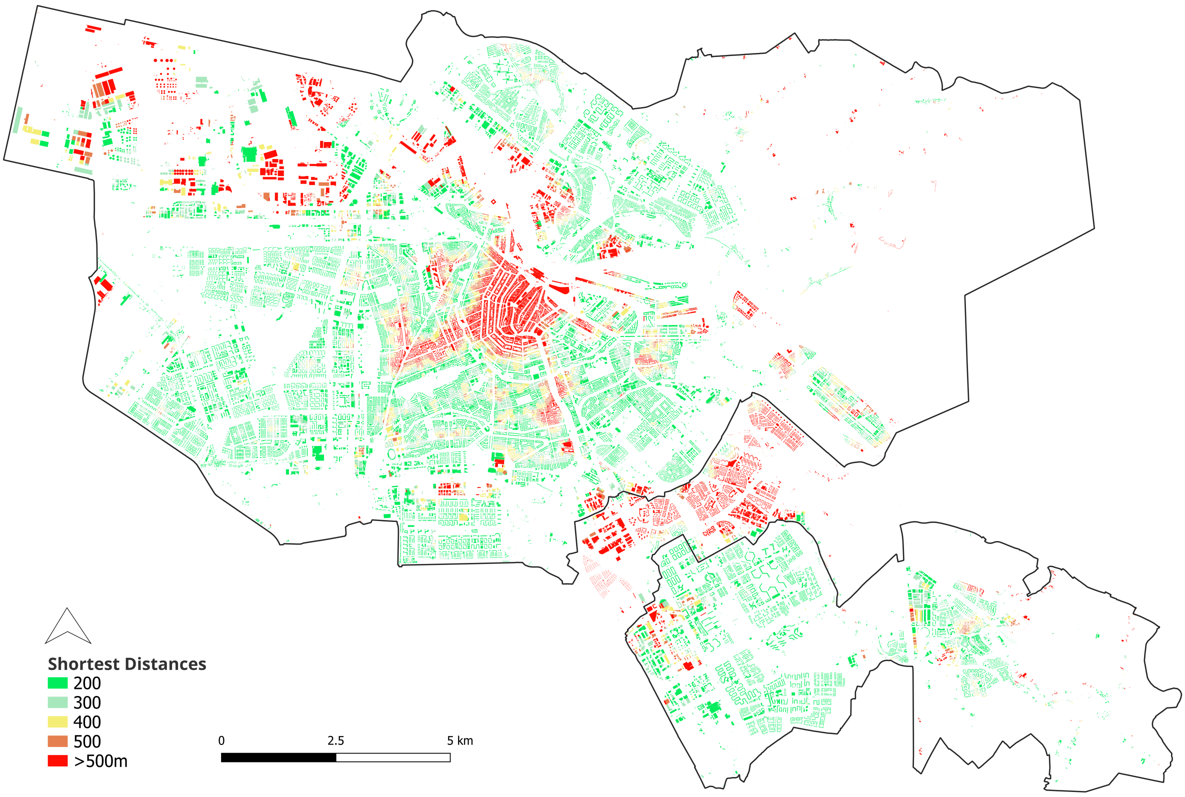



**Figure 5.17:** Walking shed map with shortest distances for the whole study area

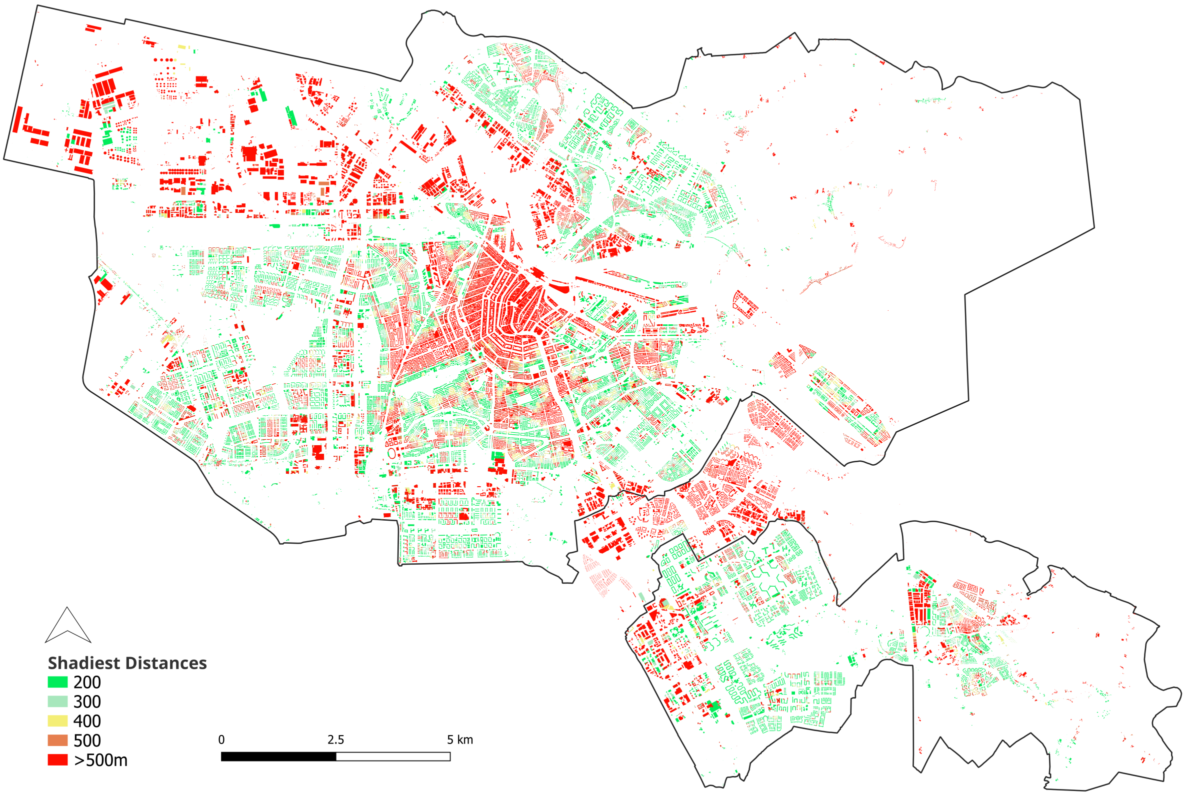



**Figure 5.18:** Walking shed map with shadiest distances for the whole study area

To assess the walking shed maps and the datasets of the assigned buildings in more detail, we decided to examine the results for 21$^{st}$ June and the time slots of 10 am, 2 pm and 6 pm as inputs focusing on the area of Vondelpark. In the following subsections, the results are presented and analyzed.

### Shortest distances to cool spaces

Looking at these three maps, it is apparent at there are no significant changes in the area north to this park during this day compared to the area south to it. At Figure 5.20, in the area south to the park we can spot that some buildings change distance categories at 2 pm, going from the "200-300" that they fell in at 10 pm to the "300-400". Similar transitions are also noticed between 2 pm and 6 pm in the same area.

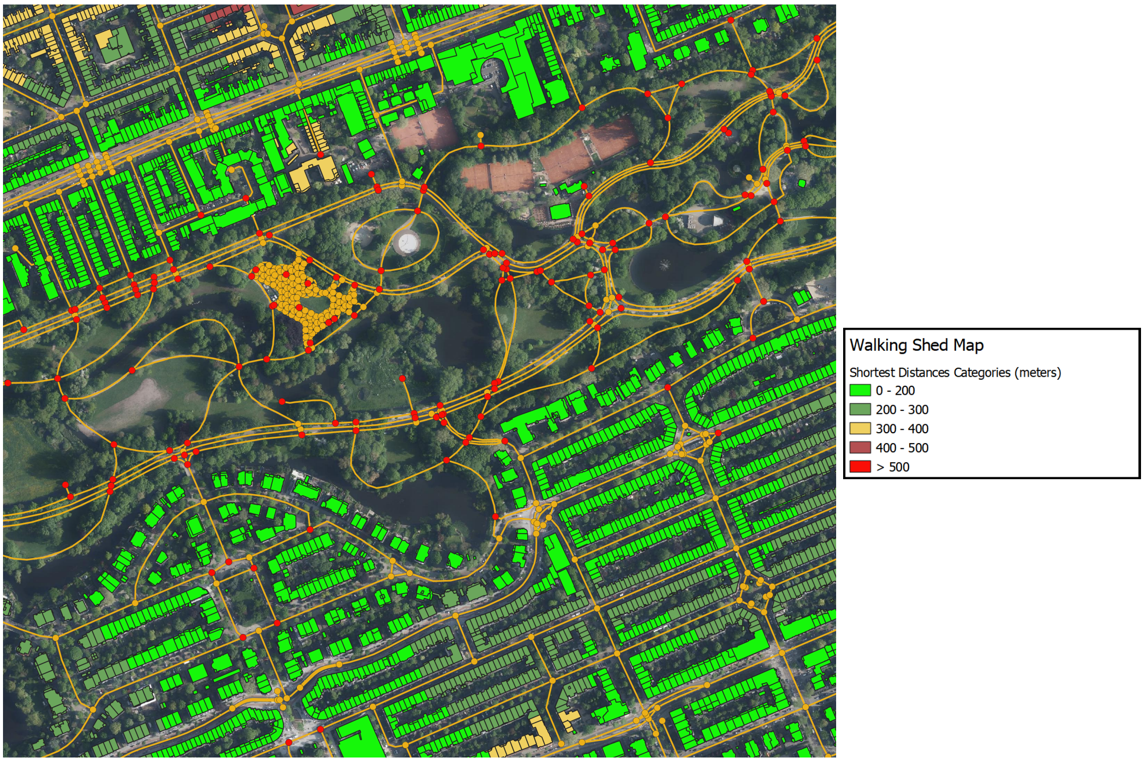



**Figure 5.19:** Walking shed map with shortest distances to cool spaces nodes -represented by red points here- on 21 June at 10 am

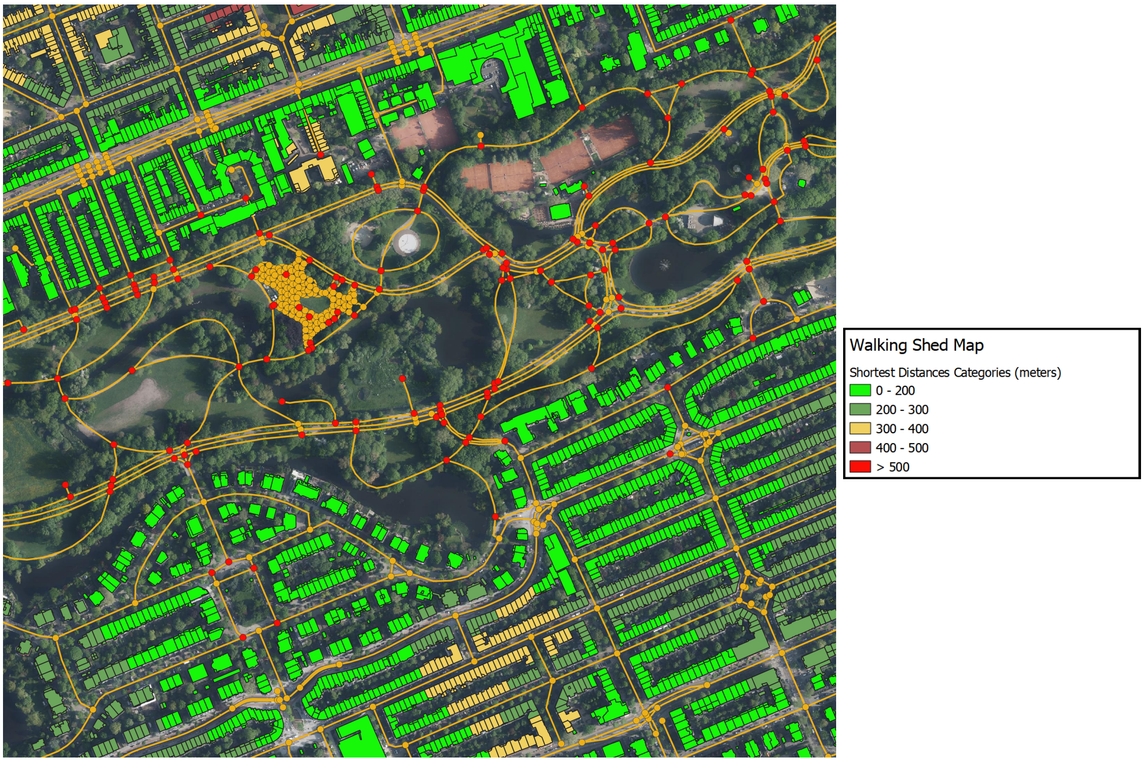



**Figure 5.20:** Walking shed map with shortest distances to cool spaces nodes -represented by red points here- on 21 June at 2 pm

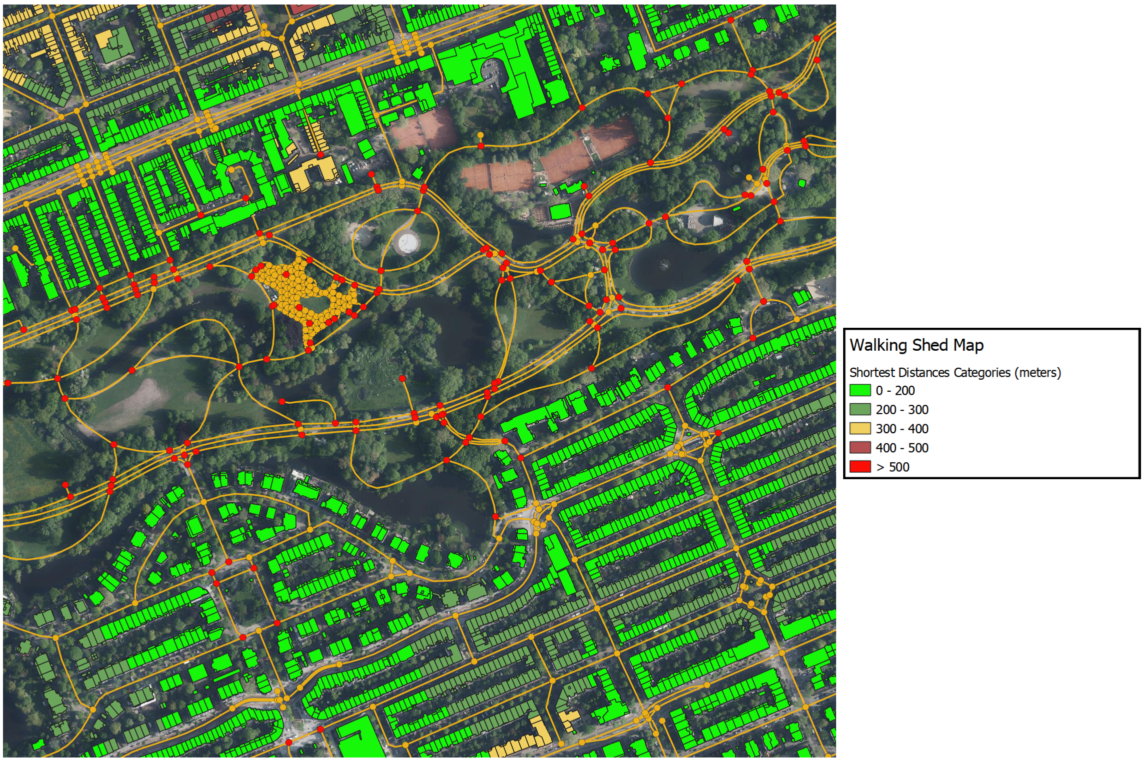



**Figure 5.21:** Walking shed map with shortest distances to cool spaces nodes -represented by red points here- on 21 June at 6 pm

Additionally it should be pointed out that although in general the results showed on these maps are considered reasonable by looking at the location of cool space nodes and buildings, some buildings north to the park are classified in categories of longer distances than expected at all the time slots. More specifically, even if these buildings have a cool space node in their vicinity, they fall in the ”300-400” category (Figure 5.22).

**Figure 5.22:** Unexpected result of walking shed maps

### Shadiest distances to cool spaces

Based on the walking shed maps that highlight distances to the shadiest routes, differences in assigned distance categories become increasingly pronounced in the area south of the park as the day progresses. The fact that there is intense transition to categories of longer distances, for instance buildings at the "200-300" at 10 am (Figure 5.23) are transferred to the "300-400" category at 2 pm (Figure 5.24) was anticipated, since at this time it is harder for shade to be found and therefore, as the distances to the nearest cool spaces, which are computed to reduce the cost of the shade weight, are longer. In addition, it is worth mentioning that at 6 pm (Figure 5.25) more apparent alterations in the same area can be noticed, as some buildings of the "> 500" at 2 pm turn into ones of either "0 - 200" or "200 - 300" category after four hours. These changes are expected, as increasing shade in the late afternoon creates shadier road segments that can be utilized to reduce the routing cost in Dijkstra's algorithm when shade is used as a weight factor.

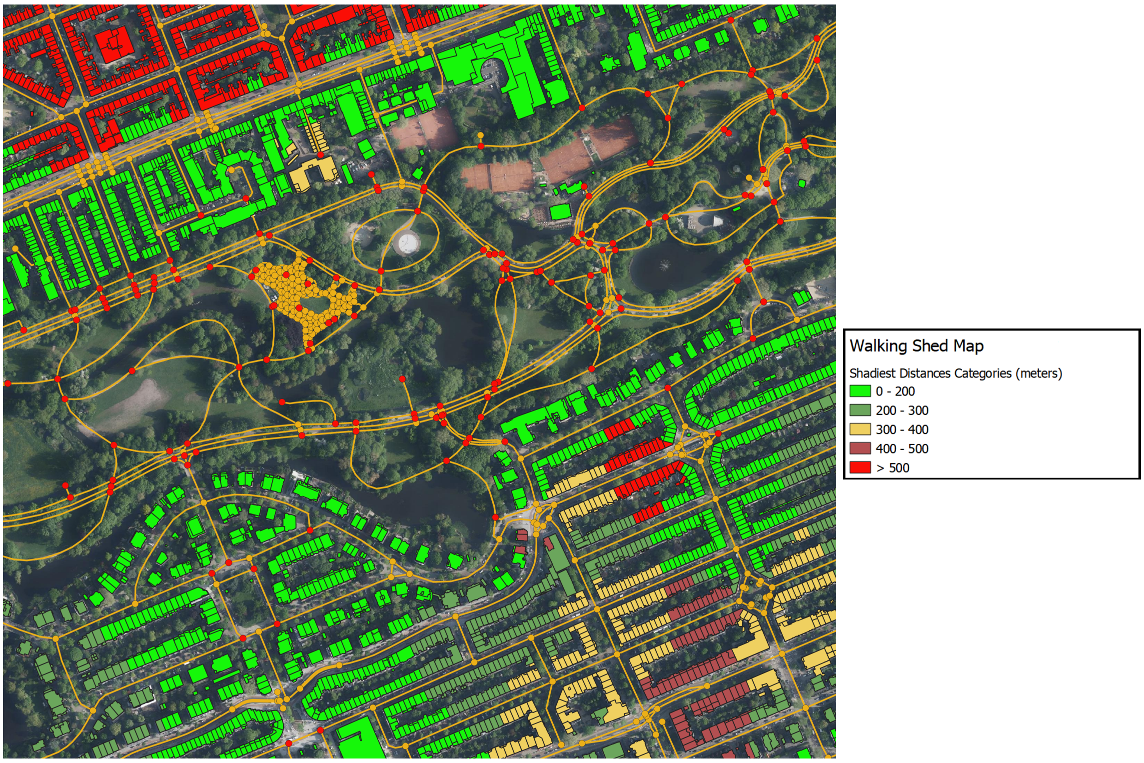



**Figure 5.23:** Walking shed map with shadiest distances to cool spaces nodes -represented by red points here- on 21 June at 10 am

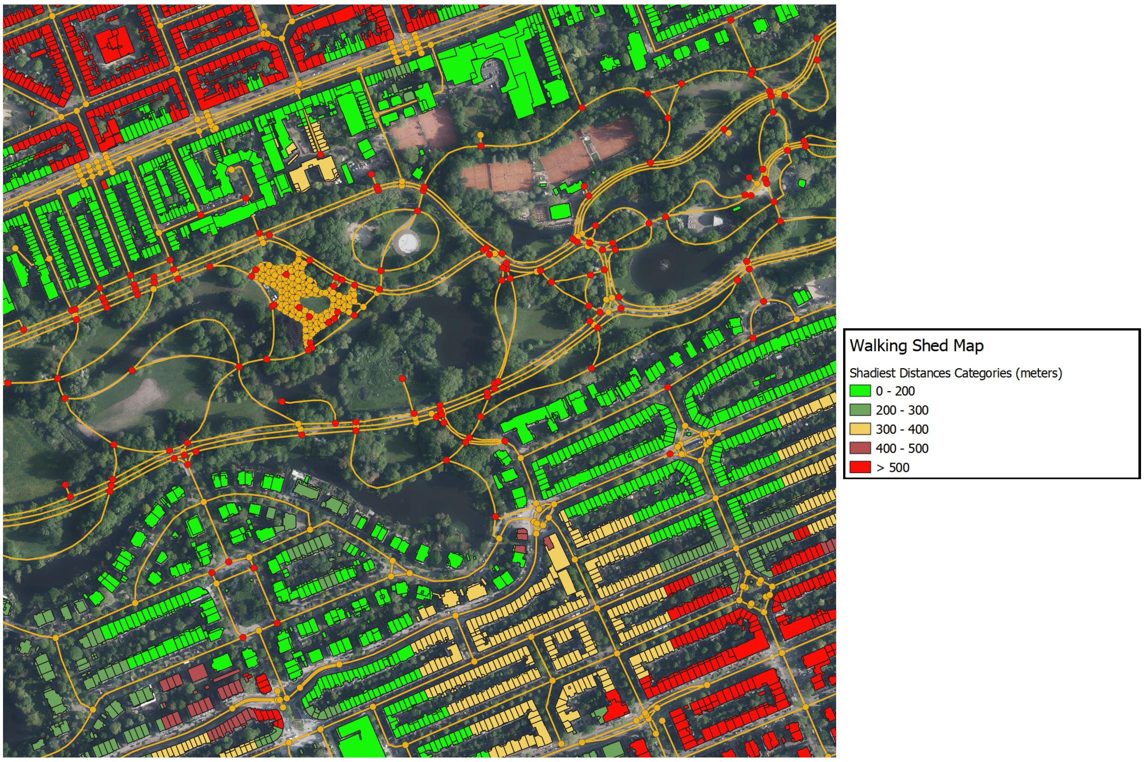



**Figure 5.24:** Walking shed map with shadiest distances to cool spaces nodes -represented by red points here- on 21 June at 2 pm

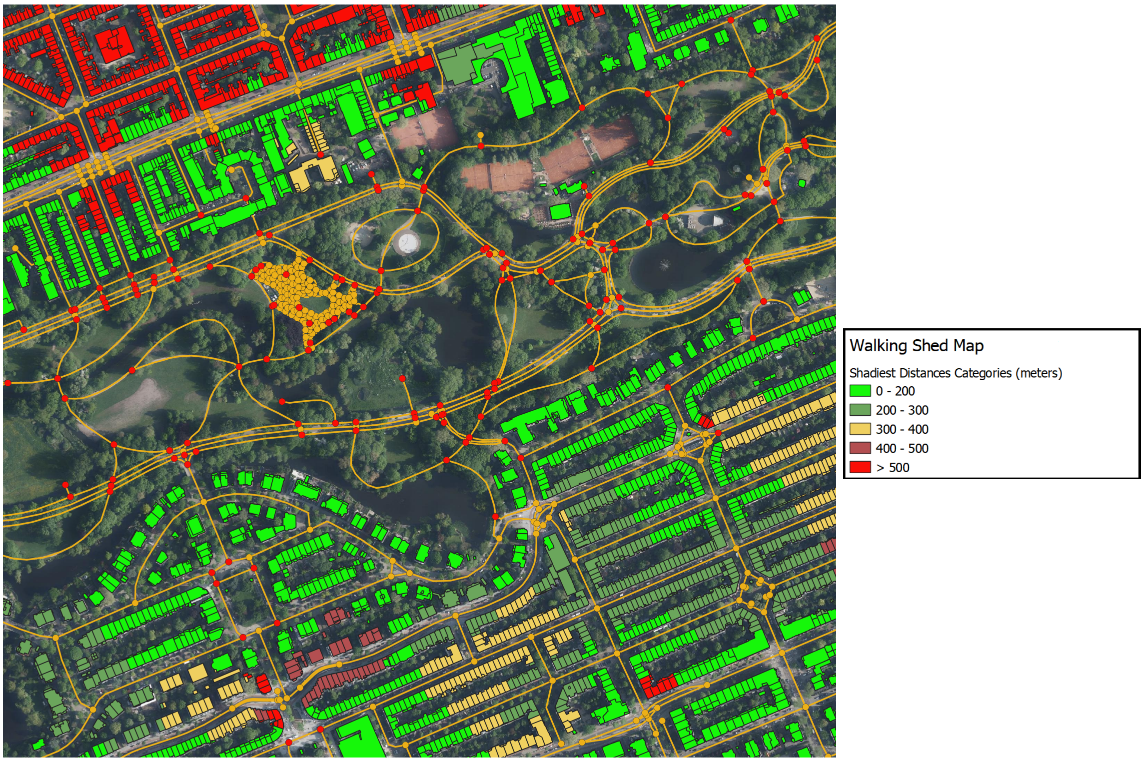



**Figure 5.25:** Walking shed map with shadiest distances to cool spaces nodes -represented by red points here- on 21 June at 6 pm

However, across all three maps, there are several abrupt ”jumps” in distance category classification, causing adjacent buildings to be placed in drastically different categories. For instance, some buildings in the ”0–200” category appear directly beside buildings classified in the opposite category, ”>500.”

This likely results from limitations in the network graph's quality, specifically due to an insufficient number of nodes representing buildings or cool spaces. In these cases, a building's centroid might be assigned to a different node on the graph, and the route from that node to the nearest cool space ends up being significantly longer than expected.

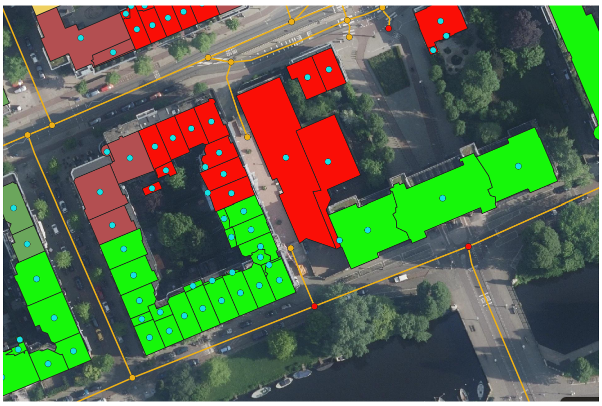


**Figure 5.26:** Incoherence in distance categories at nearby buildings caused by the lack of nodes -centroids of the buildings represented by cyan points and cool spaces nodes as red points-

### 5.3.2. Routing Algorithm

Regarding the routing algorithm, we conducted two experiments, one for each type of route, to test our implementation. The user inputs, including the method of specifying locations, the type of route and date and time, were used to guide the routing algorithm. The inputs were based on the following parameters:

- Location Specification: Coordinates
- Routing Mode: from origin to destination
- Date Time: June 21st, 2 pm

As a result, the routes were generated based on the June 21$^{st}$ dataset for 2 pm. As it can be noticed the Figures 5.27, the shadiest route is significantly longer than the others, to the extent that it surpasses the shortest one by roughly 1 km. The two balanced routes between these specific locations overlap excessively, sharing common edges that reduce routing costs in both cases. The only difference between the routes lies in the final two edges.

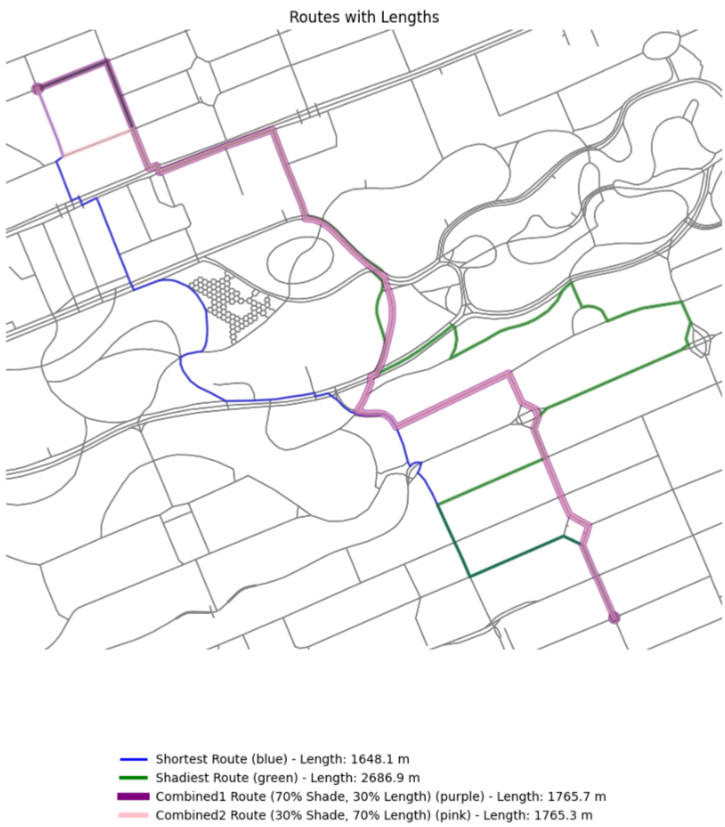



**Figure 5.27:** Origin-destination route on 21 June at 2 pm

Next, we experimented on the nearest cool space route. From Figure 5.28, we can see on the map where the origin of the route is located. Also, the corresponding parameters are presented below:

- Location Specification: Coordinates
- Routing Mode: from origin to nearest cool space
- Date Time: June 21st, 2 pm

As it can be noticed from Figure 5.29, the shortest route is almost half of the shadiest path, since the latter is calculated in this way that contains edges with greater shade weights. Additionally, in this case the balanced routes have some differences compared to the previous experiment, with the route that focuses more on the shade being longer by approximately 300 meters. Looking at the Figure 5.28, we can see on the map which the nearest cool place node is and the cool space polygon with which is related.

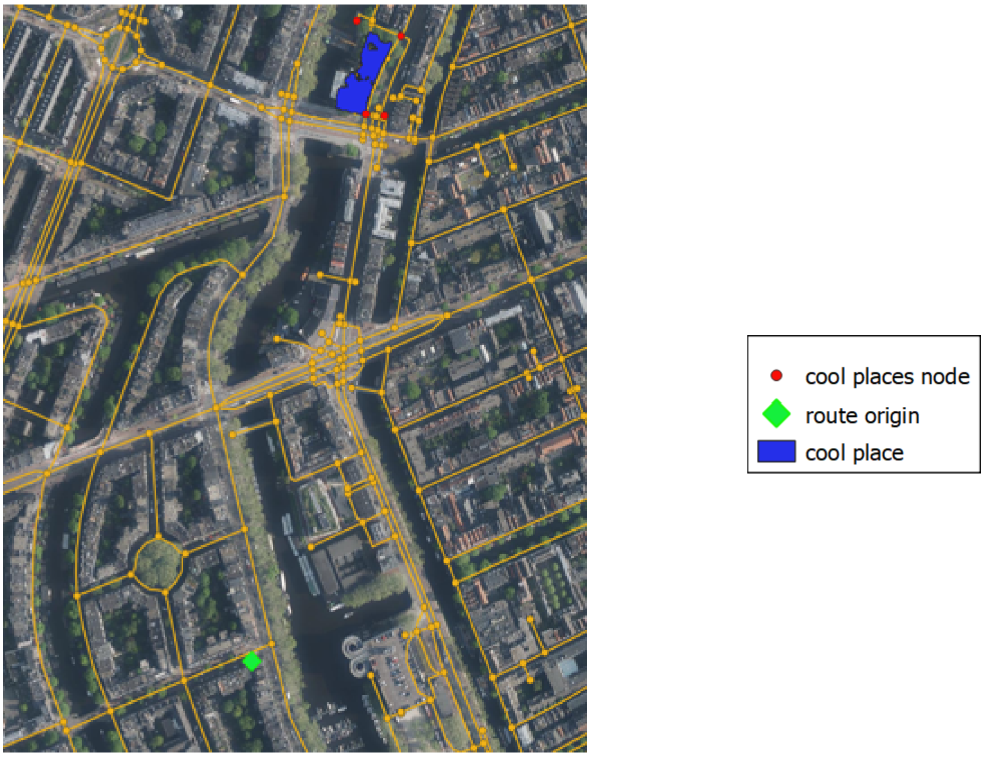



**Figure 5.28:** Origin of route, cool places and cool places nodes on the map


**Figure 5.29:** Nearest cool place route on 21 June at 2 pm

As a general note about the network graph we used, it's important to mention that some edges were missing from the OSM dataset, resulting in a disconnected network in certain areas. While this did not affect the execution of the code or our experiments, it means that if a route were requested between two locations that are not connected, it would not be possible to find a valid path.

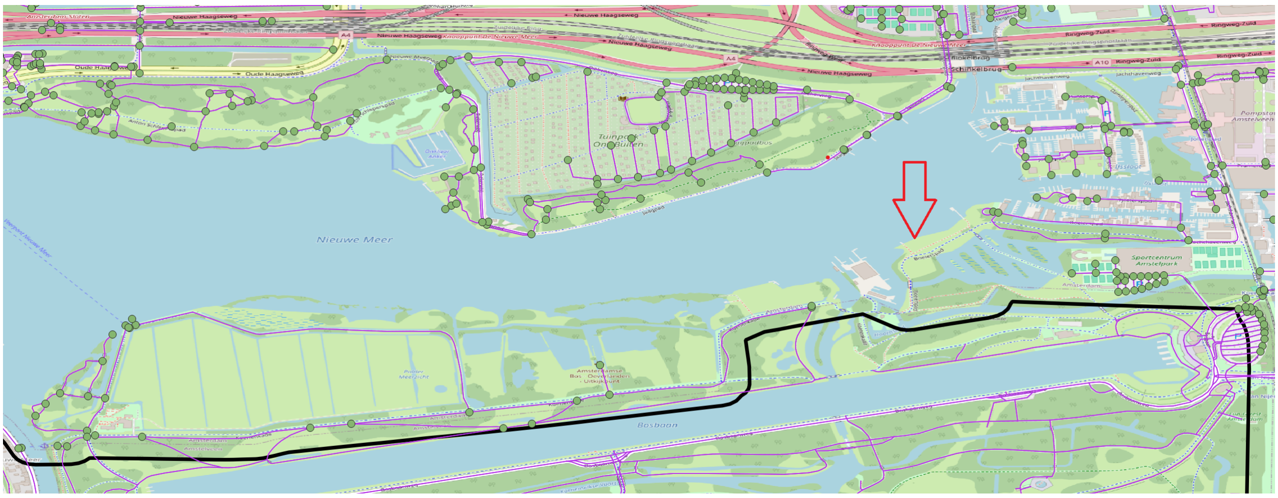


**Figure 5.30:** Example of missing edge causing disconnection

## 5.4. Comparison with the old datasets

### 5.4.1. Cool spaces

The main difference between the original and our cool space approach is that we preserve the cool space candidate polygons, instead of directly using the shade geometries. This helps avoid the 'unnatural' and sprawling shapes that can arise from converting raster data to polygons, as illustrated in Figure 5.32. Additionally, our approach allows hard surfaces, such as plazas, to be identified as cool spaces, not just green areas. This is because hard surfaces can also have shaded zones, created by surrounding human-made structures like tall buildings and urban canopies. Finally, our dataset includes a quality indication for the cool spaces

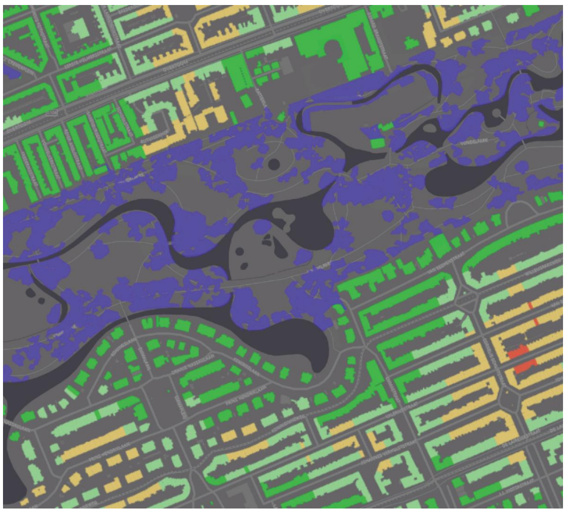


**Figure 5.31:** Original dataset of Cool spaces

**Figure 5.32:** Our Cool space result

### 5.4.2. Shortest distances

When comparing the original cool spaces dataset with the walking shed map for the shortest distances on June 21 at 2 pm, it is evident that there are many similarities, particularly with the buildings located adjacent to the park. Notably, in the area to the north of the park, the buildings with uncertain results are classified into the same distance categories as in the original dataset.

However, the fact that the cool spaces identified for this specific date and time may differ from those in the original dataset explains the discrepancies observed, particularly in the area south of the park. Additionally, the absence of sufficient network nodes impacts our results, likely hindering the accurate representation of buildings and cool spaces within the network.

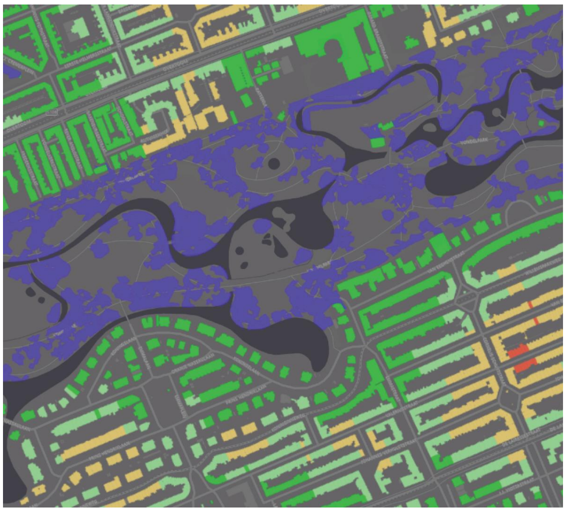


Figure 5.33: Original dataset of Cool spaces


Figure 5.34: Shortest distances on 21 June at 2 pm

### 5.4.3. Shadiest distances

A similar comparison, as described in section 5.4.2, can be made between the original cool spaces dataset and the walking shed map for the shadiest distances on June 21 at 2 pm. Again, it is observed that most of the buildings close to the park fall into distance categories that align with the original dataset. Additionally, the fact that many buildings are classified into longer distance categories compared to the original dataset is understandable, as it becomes more challenging to find shade and cool spaces at that time of day.

As discussed in section 5.3.1, the abrupt shifts in distance categories between nearby buildings were not anticipated and warrant further investigation.

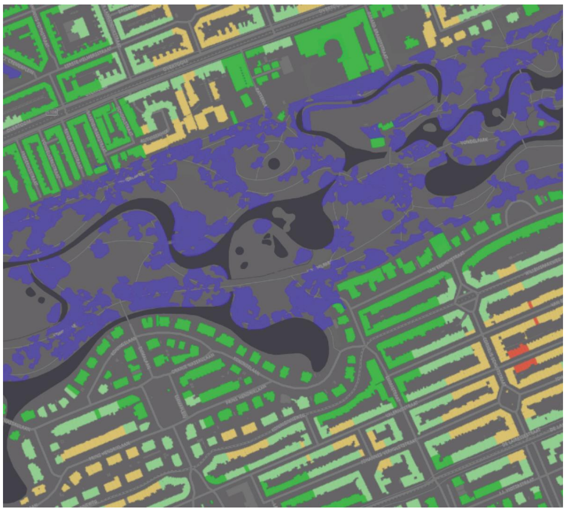


Figure 5.35: Original dataset of Cool spaces

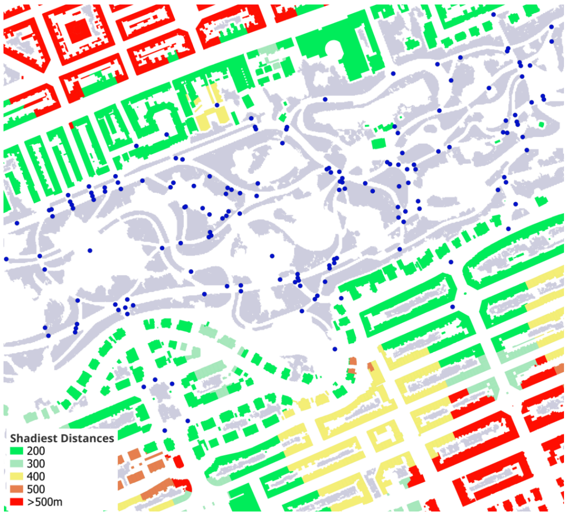



Figure 5.36: Shadiest distances on 21 June at 2 pm

# 6 Conclusions

This project aimed to provide an overview of shade and cool spaces in Amsterdam, focusing on the following three primary objectives:

1. To identify the properties which determine the quality of cool places and use these to locate and score the cool places in Amsterdam.
2. To develop a method to compute the shortest and shadiest distances for the whole of Amsterdam to cool places.
3. To develop a route-making method combining the shadiest and fastest routes, from a given location to the nearest cool space or given destination.

A fundamental requirement for achieving these objectives was the availability of data regarding shade patterns in Amsterdam. This data not only informed the shade values in the routing methods but also facilitated the identification of cool spaces. Therefore, we identified a fourth objective: to map the shade patterns of Amsterdam on representative days. Utilizing the Daily Shadow Pattern UMEP tool, we calculated shade based on solar positioning alongside DTM and CHM data. We developed a method to create the necessary CHM and DTM inputs using AHN data, assorted by GeoTiles. This involved extracting vegetation points, building heights, and ground elevations. The result was four shade maps for days in June, July, August, and September, which can be reused for earlier spring months

To address our first objective, two main steps were undertaken. First, cool spaces were identified based on criteria including public accessibility, shading, geometrical shape, and distance to roads. This process produced various shaded geometries within cool space candidates for 18 specific times of day (from 9:00 to 18:00) at 30-minute intervals, across dates from June to September. Next, the identified shaded geometries were evaluated in terms of shading, usability, capacity, heat risk, and Physiological Equivalent Temperature (PET). Finally, cool space candidates were classified based on the average score of their shaded geometries into three categories: *Highly Recommended*, *Recommended*, and *Not Recommended*. Results show that August has the highest number of *Highly Recommended* cool spaces, followed closely by July. In contrast, September has the fewest *Highly Recommended* cool spaces and the highest number of *Not Recommended* ones.

For the second and third objectives, we created an OSM network graph enriched with shade weights for each shade map. The network graph is processed by calculating shade weights based on the proportion of shaded pixels for each edge. We defined cool space nodes by overlaying the cool places onto the network graph, identifying their nearest nodes, and integrating building centroids into the network. We also defined network nodes at 200, 300, 400, and 500 meters from each cool space node and checked if building nodes fell within these distances, using both length and shade weights. This allowed us to produce walking shed maps and enrich the building shapefile with corresponding attributes. Regarding the routing algorithm, our code allows users to input routing preferences, specify locations, and determine origins and destinations. The results provide four routes from an origin to a destination or the nearest cool space.

In the end, we have created a GitHub repository containing a Python program to create the required datasets sequentially. The program can also calculate the shadiest and shortest path from given locations and the nearest cool spaces. The created datasets contain information about the shade in Amsterdam for four different dates, the location and quality of cool spaces for four different dates, and the shadiest and shortest distances from cool spaces for the whole of Amsterdam, for one day. These datasets can be used to analyze shade and cool spaces in Amsterdam, however there are some limitations to these datasets.

# 7 Reflection and Future Study

In this chapter we will reflect on our method and results, and provide recommendations for future study.

## 7.1. Reflection

We shall first reflect on our three research parts (shade, cool spaces and pedestrian network) separately, then on their integration and finally the overall research limitations.

### 7.1.1. Shade

Overall, the methodology chosen for creating the shade maps has resulted in reliable outputs for Amsterdam. The code was designed to function without requiring any preliminary data preparation from the user, which enhances convenience.

The reliance on GeoTiles for generating the input CHM and DSM presents both advantages and disadvantages. While the code is limited to use within the Netherlands and depends on GeoTiles remaining operational and updated, GeoTiles also provide convenient data for creation. The use of buffers in the subtile structure and the overall division into tiles facilitates parallel processing, enhancing efficiency. Additionally, while the 3DBAG dataset for buildings was utilized, it contained a few incorrect geometries, which for our datasets have manually be replaced with geometries from the BAG database.
Moreover, the AHN4 data was collected during the period from 2020 to 2022, with that of the Amsterdam area in 2020. As a result, it does not reflect the most current conditions in the city. While urban environments typically do not change rapidly, new buildings and infrastructure developments have occurred since then.

The primary source of inaccuracies in the shade calculations stems from the CHM creation method. The NDVI threshold retains a significant number of non-vegetation points. In an effort to create the shade maps as quickly as possible, since they are essential for the subsequent cool spaces and pedestrian network analyses, we opted for a technique previously employed in earlier research. In hindsight, it would have been beneficial to explore alternative methods, such as machine learning.

The UMEP tool code presented challenges due to its unclear parameter usage and structure. We have therefore not been able to include meaningful documentation about the specifics of this code. Although attempts were made to clean up the code, these often led to malfunctions, leaving much of it unchanged.

Finally, the creation of the input files and the shade calculation remains a time-consuming process. For the input files this is less of an issue as they typically need to be calculated only once. Implementing parallel processing improved the efficiency of shade calculations.

### 7.1.2. Cool spaces

The implementation of cool space analysis follows the structure of the methodology and gives reasonable results. Yet, some changes are made to achieve more meaningful analyses, and some limitations make the result not ideal enough.

During cool spaces identification, we have two methods for shade indicators, first is shade coverage in terms of time and shade coverage in terms of area ratio, which is important to assess whether the shade in a cool space is sufficient for people to stay for a long period of time and also for the people capacity. This adjustment allows us to produce a more comprehensive analysis of the current shade coverage situation in our research area.

In our capacity evaluation, we assigned each building individually to its nearest cool space to avoid duplication in service capacity, thereby producing a more realistic capacity analysis. However, this method has two important limitations. First, due to an insufficient number of nodes for locating cool spaces and buildings in the OpenStreetMap-based graphs, we still rely on Euclidean distance rather than actual walking distance. Second, although we managed to prevent duplicated capacity assignments, when the nearest cool space exceeds its capacity, other nearby cool spaces do not accommodate the overflow.

Besides, other processing also suffers from inadequate datasets such as Public land use identification, road type for buffer, and Street Furniture evaluation. Detailed descriptions of datasets can be found in 7.1.5. Therefore, any other solution to overcome this limitation of datasets should be taken into account. About the good results and limited results, more detailed explanations can be found in B.5.

The code development of the cool space process is not clear and flexible enough due to some hard-coded parameters such as specific attribute names used for analysis. The procedure is also not independent enough. The program only makes the identification and evaluation parts able to run separately, if an error happens during the identification part, for example, it has to be run again from the beginning.

### 7.1.3. Pedestrian Network

For the pedestrian network, we have managed to develop a workflow for the user which produces meaningful datasets and a routing algorithm tailored to our project's needs. However, there is room for improvement.

Initially, we effectively addressed the challenge of integrating OSM data, open geospatial data provided by the Municipality of Amsterdam, and information derived from the shade map creation and cool spaces analysis. This integration allowed us to generate comprehensive network graphs of Amsterdam that incorporate all relevant geospatial data from our research. Additionally, our executable can be applied to other cities, provided that the necessary network graphs, building polygons, shade maps, and cool spaces polygons for the target area are available for the specified date and time.

Nevertheless, it was noticed that the OpenStreetMap network had missing edges in some parts of the Amsterdam, especially in areas outside the dense urban environment and, as mentioned previously, its edges were not enough to represent sufficiently the external entities used for our analysis, such as buildings and cool spaces polygons. Consequently, if time allowed it, more attempts would be made in order to find ways to supplement the network graph accurately by defining more representative nodes for buildings and cool spaces.

In addition, with our current executable, the creation of the walking shed with the shadiest distances is quite slow, as the computations made for this map are more complicated compared to the map with the shortest distances. In case we had more time at our disposal, we would manage to optimize it.

Finally, the walking shed maps reveal that some neighboring buildings exhibit different walking distances, with values jumping from 200 to 400 meters. This discrepancy may be attributed to the buildings being assigned different nearest nodes within the pedestrian network. In addition, since the time was limited for the cool space computation to be further tested, there is a possibility that it accounts for the unexpected results in walking shed maps.

### 7.1.4. Integration of the three research parts

A single main code file has been developed to run all three research parts in sequence, adhering to their dependencies: "cool spaces" requires the "shade map" data as input, while "pedestrian network" depends on outputs from both the "shade map" and "cool spaces" data. Each part is designed to directly use the output data from previous steps. Together the parts create the requested datasets.

However, the overall integration of these components lacks cohesion. Differences exist across the three parts in configuration file setups, naming conventions, code structures, and the structure of sections in both the report and the GitHub documentation. Ideally, a standard would have been made for all of these elements. Our focus on developing each research part individually, however, led to a lack of uniformity, reducing the code's ease of use.

Furthermore, there were miscommunications regarding the shade database, stemming from the expectation that the routing component would have a pre-made database for specific dates, enabling users to easily calculate a route in Amsterdam. This included the anticipated reuse of shade maps for the before and after solstice dates, but this feature was not developed. Additionally, there was a miscommunication about the dataset of July 1st 2015 being the representative day for the outputs, as this is the same date for which the PET map was calculated. This led to the use of the June 21st data for Amsterdam instead.

Additionally, running the code for all three parts requires manual input of data paths from one part to another in the configuration files, as these paths are not generated automatically.

### 7.1.5. Research limitations

Our research limitations primarily involve the usability of our code when data updates occur and data sources change, and the quality of the datasets used.

The current framework for shade map generation is dependent on GeoTiles and AHN data, creating a limitation as it is not designed to accommodate alternative resources. This dependency poses a risk, particularly if GeoTiles are no longer updated with future AHN data, which would mean a new methodology for generating the input data for shade maps has to be created. Additionally, the process requires that the subtiles have an overlapping buffer, which adds complexity should alternative data sources be considered, as overlapping buffers would need to be created for any new datasets. Finally, this data source only covers the Netherlands, and therefore limits the usage of our code for analysis of other countries.

For the cool space process, two main challenges are the insufficient quality of datasets and the lack of references for classification.

Regarding the quality of datasets, the PET map is only made for representative hot days, thus it is not accurate enough for the less warm days. Besides, the PET map is the result of 2015, which is incompatible with the shade maps which represent the year 2023. The street furniture data is not complete, which results in insufficient analysis of the missing data areas. The road data does not clearly distinguish motorized and non-motorized roads, which makes the buffer setting hard to be precise. Finally, the land use data has both geometrical shape and classification problems. For the geometrical shape problem, a non-ideal division of cool spaces results in an incorrect capacity indicator computation mentioned in section 7.1.2. For the classification problem, the current classification cannot distinguish some of the irrelevant spaces such as parking lots beside buildings, this results in some incorrect recommendations in the final output.

For classification, we lack solid references, thus we can only set up standards from the gut. For example, when setting the distance buffer based on road types, we simply decided to give a 20-meter buffer for highways, a 15-meter buffer for regional roads and so on. When setting weights for different indicators, there are also no references, we set the weight of indicators based on our perceptions, which makes the final recommendation less scientific, more subjective.

As far as the pedestrian network analysis part is considered, firstly our limitations are related to the OSM network, since some edges are missing in specific areas in Amsterdam. This issue is less pronounced in our study area, because it mainly covers a dense urban environment, where OSM network is more extensive. However, the full coverage of walkable network in Amsterdam would be the preferable. In addition, the nodes in the graph are not as adequate we expected to represent the cool space nodes and the buildings, something inevitably affects the results especially for the walking shed maps, where the network graph is not so dense. Also, due to limited time for further testing of the cool space computation, apart from the graph network it may have also contributed to the unexpected results in the walking shed maps. Finally, the built-in functions of Networkx do not provide a direct way to find weighted path lengths according to Dijkstra's algorithm where the weight differs from length but where the cut-off distance (length-sum of edge weights-at which the search is terminated) is still based on length. Although our implementation was effective, the computational time was substantial.

## 7.2. Future Study

Future studies can focus on integrating more sources for CHM and DSM creation and developing a better CHM creation method, creating data for cool space evaluation, creating a better pedestrian network, and analyzing and visualizing our results.

As previously discussed, relying solely on GeoTiles for CHM and DSM creation may pose challenges in the future, particularly as it limits applicability to the Netherlands. To address this, additional data sources could be integrated, such as PDOK (Public Data Key) for the Netherlands and open data from other countries. By incorporating a country input parameter into the code, the system could dynamically access datasets. This integration would necessitate the development of a custom tiling system to manage the data and enable parallel processing.

Moreover, to enhance the quality of the CHM, various CHM creation methods can be tested and compared. By exploring diverse techniques, such as advanced photogrammetry and machine learning approaches, future studies could yield more accurate and reliable models.

Due to time constraints, this project currently relies on secondary datasets for assessing cool space indicators, which significantly limits the scope of analysis based on data availability. Future studies should consider incorporating primary data collection methods, such as on-site observation, to allow a broader range of indicators to be evaluated. Cool spaces can be seen not only as areas where people can cool down during travel but also as destinations for socializing and recreation. Thus, aspects of cool spaces such as landscape design, biodiversity, and amenities (e.g., fountains, playgrounds, skate parks) play an essential role in their attractiveness and can encourage visitation. Future research should further elaborate on these elements, along with inclusiveness, comfort, and safety, aligning these indicators with the SDGs, as can be adapted from UN Habitat indicators(Scruggs, 2020). Additionally, as this project will provide walking directions to cool spaces for users, incorporating a user review feature could enhance the tool by allowing users to evaluate cool spaces based on their experiences, providing valuable insights for other users. Beyond user-based assessments, automatic assessments could also be integrated using Street View Imagery and Computer Vision techniques (Chen & Biljecki, 2023).

In regards to the pedestrian network, in a future work it could be examined how network graph can be of higher quality for this kind of applications, especially in the areas on the outskirts of a city of rural areas. Although adding nodes and edges can be achieved with package like Networkx, the accurate way to do this should be further investigated. Additionally, further testing on the cool space nodes calculation and route computation, especially the shadiest one can be done. Moreover, the routing algorithm could be refined so that it includes reuse of shade maps for the before and after solstice dates instead of selecting the closest date and time to the user's input compared to the dates and time slots in the dataset. Lastly, it can further examined how processing time of the dataset for the walking shed map with shadiest distances can be reduced.

In general, our project has focused primarily on creating the datasets rather than visualizing and analyzing the results. Currently users must manually import the resulting geodata files into visualization software. Developing an application that automatically displays and styles the data upon completion, would be a valuable addition. A true application for calculating the routes and presenting them as a map, instead of a plot would also be meaningful. Thereby, for future studies the variations in shade across different days and times can be analyzed, as well as how these variations influence cool spaces and routing options. This analysis could yield insights into the dynamic nature of urban shade patterns.

# References


AHN. (2020). Kwaliteitsbeschrijving [Accessed: 2024-10-28]. https://www.ahn.nl/kwaliteitsbeschrijving

Aleksandrowicz, O. (2022). Chapter 1 - mapping and management of urban shade assets: A novel approach for promoting climatic urban action. In A. Khan, H. Akbari, F. Fiorito, S. Mithun, & D. Niyogi (Eds.), *Global urban heat island mitigation* (pp. 1–27). Elsevier. https://doi.org/https://doi.org/10.1016/B978-0-323-85539-6.00003-2

Aleksandrowicz, O., Zur, S., Lebendiger, Y., & Lerman, Y. (2020). Shade maps for prioritizing municipal microclimatic action in hot climates: Learning from tel aviv-yafo. *Sustainable Cities and Society*, *53*, 101931. https://doi.org/https://doi.org/10.1016/j.scs.2019.101931

Chen, S., & Biljecki, F. (2023). Automatic assessment of public open spaces using street view imagery. *Cities*, *137*, 104329.

Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, *1*, 269–271. http://eudml.org/doc/131436

Du, J., Liu, L., Chen, X., & Liu, J. (2020). Field assessment of neighboring building and tree shading effects on the 3d radiant environment and human thermal comfort in summer within urban settlements in northeast china. *Advances in Meteorology*, *2020*(1), 8843676. https://doi.org/https://doi.org/10.1155/2020/8843676

Gemeente Amsterdam. (2024). Hitte klimaatstress - amsterdam klimaatportaal [Accessed: 2024-10-30]. https://experience.arcgis.com/experience/aec3fc23a4134a3da32c7cb2cf3b66da/page/Hitte---Klimaatstress/

Gonzalez, R. C., & Woods, R. E. (2018). 3.6 - sharpening (highpass) spatial filters. In *Digital image processing (4th edition)* (pp. 174–175). Pearson.

Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.*, *4*, 100–107. https://api.semanticscholar.org/CorpusID:206799161

Hashim, H., Abd Latif, Z., & Adnan, N. (2019). Urban vegetation classification with ndvi threshold value method with very high resolution (vhr) pleiades imagery. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLII-4/W16*, 237–240. https://doi.org/10.5194/isprs-archives-XLII-4-W16-237-2019

Hua & Abdullah. (2018). Weighted sum-dijkstra's algorithm in best path identification based on multiple criteria. *Journal of Computer Science Computational Mathematics*, *8*(3), 123–456.

HvA, H. v. A. (2022). *Coolkit*. Hogeschool van Amsterdam.

Jamei, E., Rajagopalan, P., Seyedmahmoudian, M., & Jamei, Y. (2016). Review on the impact of urban geometry and pedestrian level greening on outdoor thermal comfort. *Renewable and Sustainable Energy Reviews*, *54*, 1002–1017. https://doi.org/https://doi.org/10.1016/j.rser.2015.10.104

Konarska, J., Lindberg, F., Larsson, A., Thorsson, S., & Holmer, B. (2013). Transmissivity of solar radiation through crowns of single urban trees—application for outdoor thermal comfort modelling. *Theoretical and Applied Climatology*, *117*(3–4), 363–376. https://doi.org/10.1007/s00704-013-1000-3

Koninklijk Nederlands Meteorologisch Instituut (KNMI). (2016). Uitleg over Warme Dagen [[Accessed: 2024-09-23]]. https://www.knmi.nl/kennis-en-datacentrum/uitleg/warme-dagen

Laan, C. M., & Piersma, N. (2021). Accessibility of green areas for local residents. *Environmental and Sustainability Indicators*, *10*, 100114. https://doi.org/https://doi.org/10.1016/j.indic.2021.100114

Ledoux, H., Ohori, K. A., Peters, R., & Pronk, M. (2023). Spatial interpolation: Deterministic methods [Accessed: 2023-11-12]. In *Computational modelling of terrains v2023.0*. TU Delft. https://github.com/tudelft3d/terrainbook/releases

Lee, I., Voogt, J. A., & Gillespie, T. J. (2018). Analysis and comparison of shading strategies to increase human thermal comfort in urban areas. *Atmosphere*, *9*(3). https://doi.org/10.3390/atmos9030091

Lin, T.-P., Matzarakis, A., & Hwang, R.-L. (2010). Shading effect on long-term outdoor thermal comfort. *Building and environment*, *45*(1), 213–221.

Lindberg, F., & Grimmond, C. S. B. (2011). The influence of vegetation and building morphology on shadow patterns and mean radiant temperatures in urban areas: Model development and evaluation. *Theoretical and Applied Climatology*, *105*(3–4), 311–323. https://doi.org/10.1007/s00704-010-0382-8

Middel, A., Selover, N., Hagen, B., & Chhetri, N. (2016). Impact of shade on outdoor thermal comfort—a seasonal field study in tempe, arizona. *International Journal of Biometeorology*, *60*(12), 1849–1861. https://doi.org/10.1007/s00484-016-1172-5

Mumcu, S., & Yılmaz, S. (2016, October). Seating furniture in open spaces and their contribution to the social life.

NASA. (2000). Measuring vegetation (ndvi & evi [Accessed 28-10-2024]. https://earthobservatory.nasa.gov/features/MeasuringVegetation/measuring_vegetation_2.php

Nave, C. R. (2004). The ecliptic and the seasons [Accessed: 2024-10-29]. http://hyperphysics.phy-astr.gsu.edu/hbase/eclip.html

Ratti, C., & Richens, P. (1999). Urban texture analysis with image processing techniques. In G. Augenbroe & C. Eastman (Eds.), *Computers in building: Proceedings of the caadfutures'99 conference. proceedings of the eighth international conference on computer aided architectural design futures held at georgia institute of technology, atlanta, georgia, usa on june 7–8, 1999* (pp. 49–64). Springer US. https://doi.org/10.1007/978-1-4615-5047-1_4

Scruggs, G. (2020). *Public space site-specific assessment: Guidelines to achieve quality public spaces at neighbourhood level*. UN-Habitat.

Steeneveld, G. J., Koopmans, S., Heusinkveld, B. G., van Hove, L. W. A., & Holtslag, A. A. M. (2011). Quantifying urban heat island effects and human comfort for cities of variable size and urban morphology in the netherlands. *Journal of Geophysical Research: Atmospheres*, *116*(D20). https://doi.org/https://doi.org/10.1029/2011JD015988

Terrani, E., Picción, A., Bentancur, O., & Cruz, G. (2024). Effect of street trees shade on perceived thermal comfort in a south temperate climate: The sidewalks of montevideo (uruguay). *Heliyon*, *10*(12).

UMEP. (n.d.). Solar radiation daily shadow pattern [Accessed: 2024-10-21]. https://umep-docs.readthedocs.io/en/latest/processor/Solar%20Radiation%20Daily%20Shadow%20Pattern.html

# A
# Task Division

**Table A.1:** Distribution of the workload

| | Responsibilities in the report | Role(s) performed |
|---|---|---|
| Jessica | 2, 3.1.1, 4.1, 5.1, 6, 7, B.2, | project manager, editor, shading code |
| Victoria | 3.1.3, 7, B.1 | communication manager, pedestrian network code |
| Haohua | 4.2, 5.2.1, B.4, B.5, 7 | technical manager, cool space code |
| Yan | 4.3, 5.3 | technical manager, pedestrian network code |
| Citra | 1, 3.1.2, 5.2.2 | data manager, cool space code |

# B
# Supplement

## B.1. Python code to extract the network graph of study area

```
# File paths
nodes_file = "C:/AMS/nodes_buffered.shp"
output_path = "C:/Github_synthesis/shady_amsterdam/PedestrianNetwork/StudyArea.graphml"

# Load the base graph for the Metropolitan Region Amsterdam
G = ox.graph_from_place("Metropolitan Region Amsterdam", network_type='walk')

# Load the custom nodes shapefile into a GeoDataFrame
nodes_gdf = gpd.read_file(nodes_file)

# Create a new graph for custom nodes from the shapefile
G_study_area = nx.Graph()
for idx, row in nodes_gdf.iterrows():
    node_id = row['osmid']  # Adjust if the 'osmid' column name differs
    x, y = row.geometry.x, row.geometry.y
    G_study_area.add_node(node_id, pos=(x, y))

# Find common nodes between G and G_study_area
G_node_ids = set(G.nodes())
G_study_area_node_ids = set(G_study_area.nodes())
common_node_ids = G_node_ids.intersection(G_study_area_node_ids)

print(f"Number of common nodes: {len(common_node_ids)}")
if not common_node_ids:
    print("No common nodes found.")

# Create a subgraph of G based only on the common nodes with G_study_area
G_subgraph = G.subgraph(common_node_ids).copy()

# Ensure the CRS is set correctly in the subgraph for compatibility with OSMnx
G_subgraph.graph['crs'] = G.graph.get('crs', 'epsg:4326')

# Check if the subgraph has edges
print(f"G_subgraph has {G_subgraph.number_of_nodes()} nodes
and {G_subgraph.number_of_edges()} edges.")
if G_subgraph.number_of_edges() == 0:
    print("No edges in G_subgraph. This could indicate that the nodes are not connected.")
```

```
# Remove any isolated nodes (those without edges)
isolated_nodes = list(nx.isolates(G_subgraph))
G_subgraph.remove_nodes_from(isolated_nodes)

# Convert the subgraph to a MultiDiGraph to ensure compatibility with OSMnx
G_subgraph = nx.MultiDiGraph(G_subgraph)

# Save the cleaned subgraph to GraphML
ox.save_graphml(G_subgraph, filepath=output_path)
print("Disconnected nodes removed, and graph saved to:", output_path)
```

## B.2. Extra information shade map database creation

This section contains more in depth information about specific settings in the creation of shade maps.

### B.2.1. NDVI filtering

The NDVI threshold selection represents a balance between capturing the most vegetation points possible and minimizing non-vegetation points. We further examined thresholds within the 0.0–0.1 range. While an NDVI of 0.1 captures primarily true vegetation points, it leads to the exclusion of some smaller trees and results in partial canopy coverage, with some side portions missing. On the other hand, an NDVI of 0.0 includes fuller canopies and smaller trees but introduces a significant number of non-vegetation points.

In figure B.3 we observe an urban area near a highway and adjacent train tracks. A lower threshold (image b) introduces additional points on the highway (upper section) and the train tracks (the smaller path below). In detail (Figure B.3) the lower threshold makes overhead structures on the highway visible which are not vegetation (above), though it also captures more trees (below).

Figure B.1 displays a dense urban area featuring a park. With a lower threshold, courtyard trees appear fuller. In detail (Figure B.2), the lower threshold results in a larger, incorrectly classified high area (upper section), though it also includes fuller, smaller trees (lower section).

We have ultimately chosen a lower NDVI value of 0.0, aiming to capture the fullest canopy coverage and to ensure we include smaller street trees that provide some shade. Urban objects such as cars, curbs, barriers, and other features are primarily filtered out by setting a minimum vegetation height of 2 meters. The highest misclasified points are filtered out by setting a maximum height for vegetation, based on the tallest possible vegetation in the area. Higher misclassified points near buildings that fall within the minimum and maximum range mostly cast shadows that will be covered by actual building shadows in the final shade maps. We observed that in some areas these points create fake shade on rooftops, but rooftops are not considered in this research. Lastly, thin horizontal objects at higher elevations—such as structures for overhead lines at train tracks or highway information boards—can create relatively large inaccurate shade in the shade maps. However, since these objects are generally located around areas inaccessible to pedestrians, we accept these limitations in our analysis.

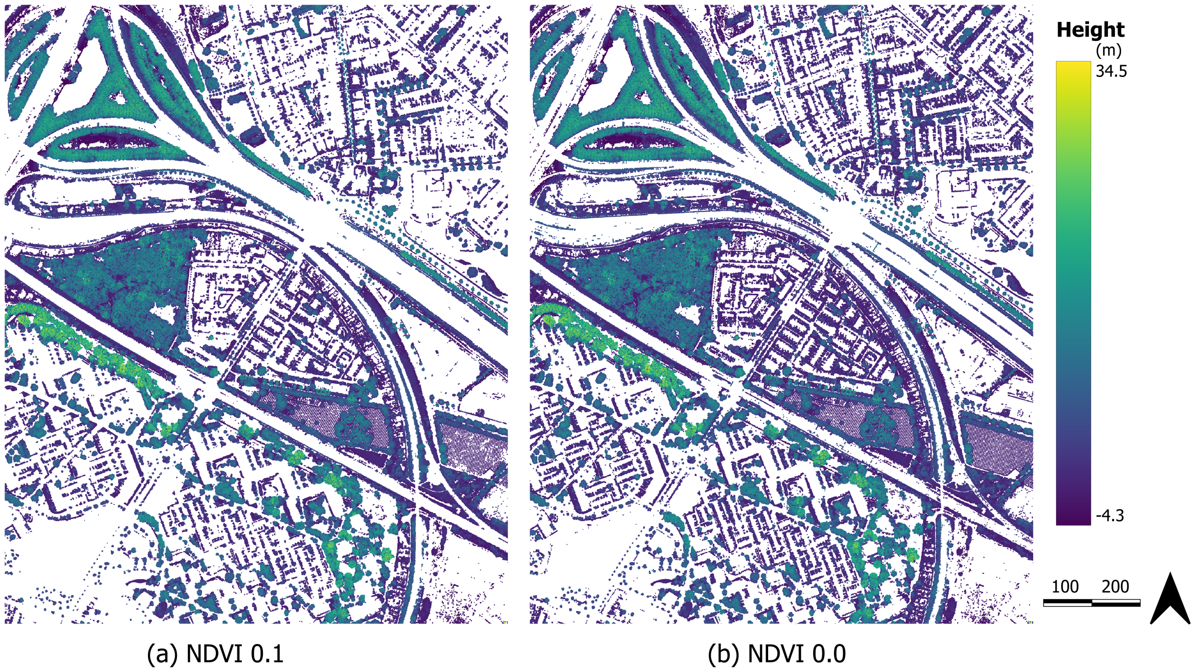



**Figure B.1:** CHM of subtile 25GN2-12 with different NDVI thresholds.

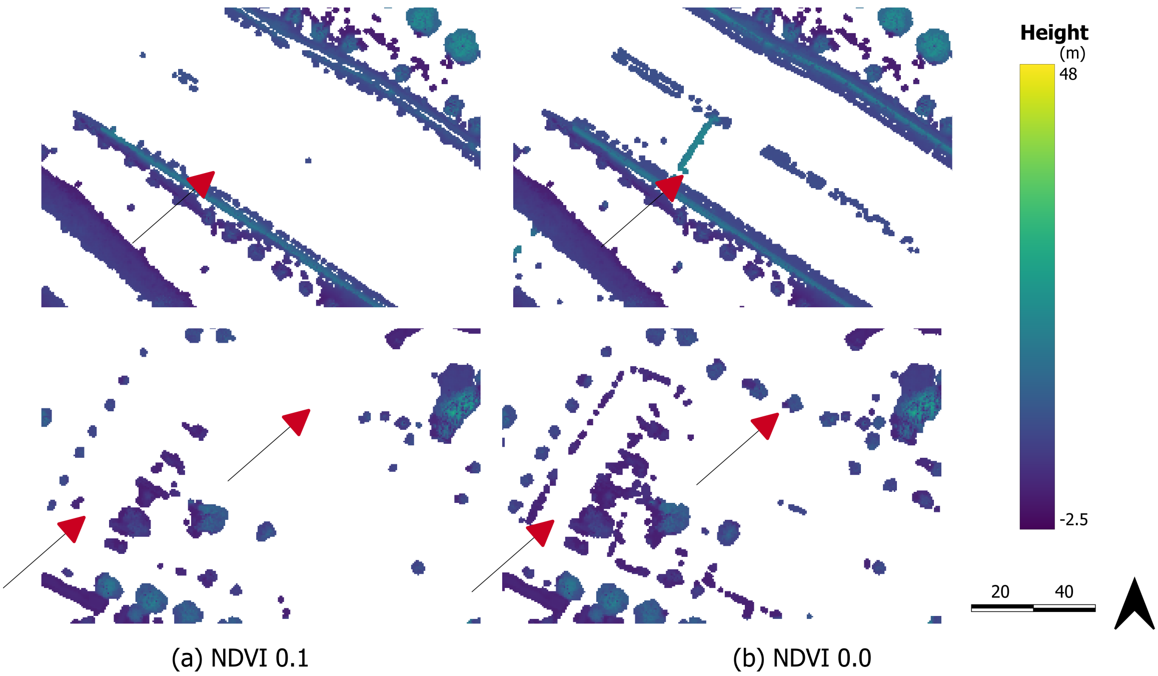



**Figure B.2:** CHM of subtile 25GN2-12 with different NDVI thresholds in detail.

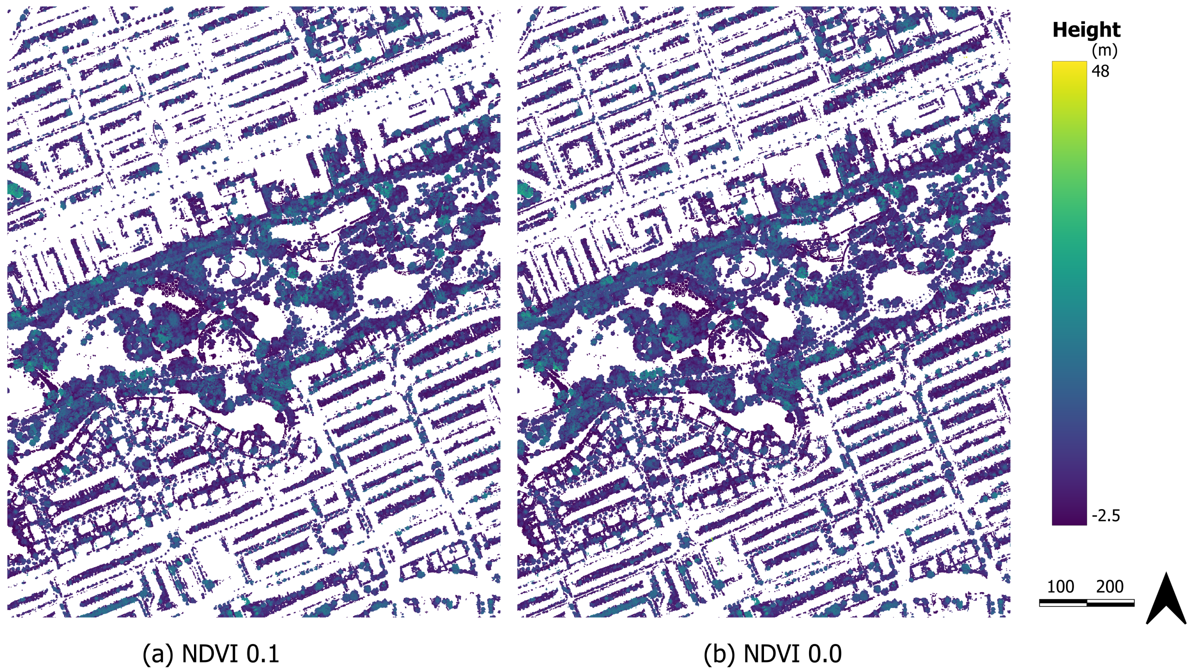



**Figure B.3:** CHM of subtile 25DN2-10 with different NDVI thresholds.

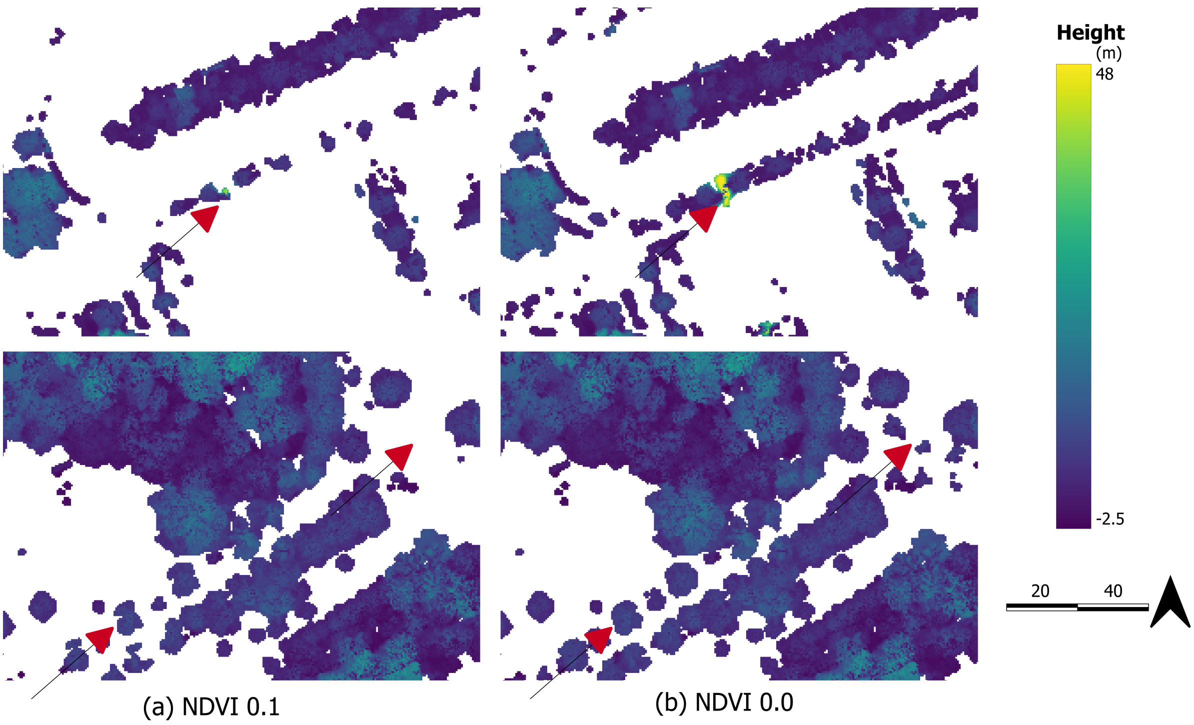



**Figure B.4:** CHM of subtile 25DN2-10 with different NDVI thresholds in detail.

## B.2.2. CHM smoothing

During the point cloud acquisition lidar collected, laser can travel through canopies and hit lower vegetation points or the ground. These points introduce 'pits' into our CHM, small gaps with lower height in the canopies.

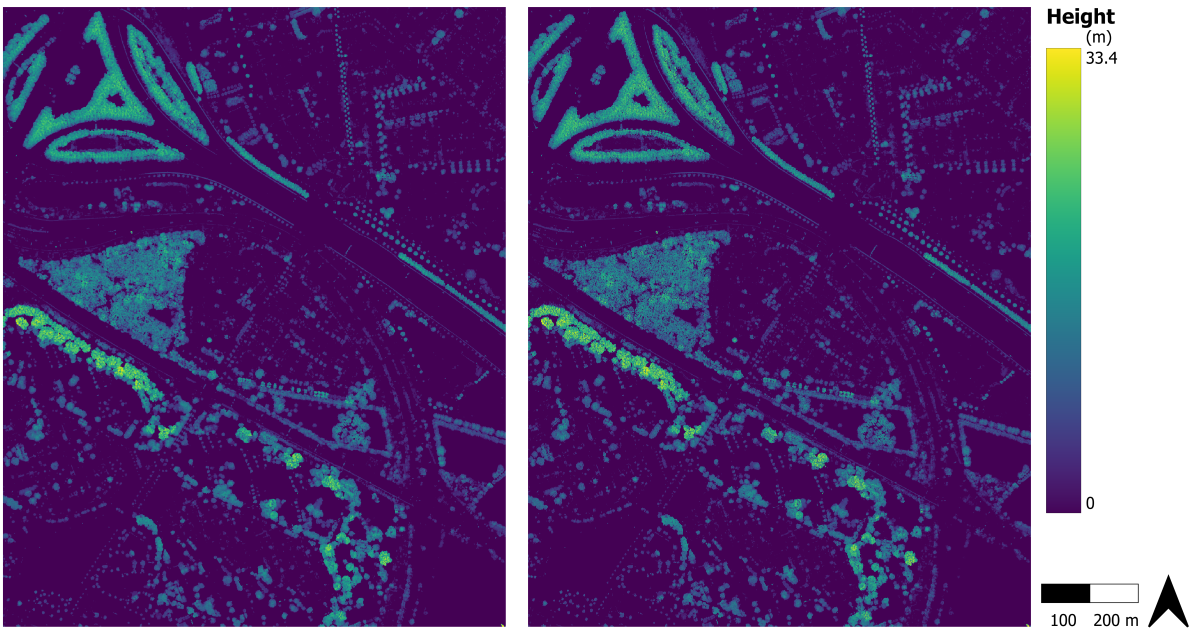
LiDAR data acquisition in forested areas often results in the laser pulse penetrating through canopy layers, reaching lower vegetation or even the ground. This penetration leads to ”pits,” or small gaps, in the CHM, which appear as points of unexpectedly low height in the canopy layer. Such gaps can interfere with accurate CHM analysis, as they create an uneven representation of canopy height and continuity.

One method for addressing these gaps is applying a median filter. A median filter is a non-linear digital filtering technique widely used in image processing, particularly in handling ”salt-and-pepper” noise. It works by replacing each pixel’s value with the median value from its neighboring pixels within a defined radius (Gonzalez & Woods, 2018). This approach preserves canopy structure by maintaining edges while removing isolated low points, effectively smoothing out pits without blurring the canopy’s natural texture. We have applied a median filter with a footprint of three, meaning the nine neighbors of a point are used.

Figure B.5 demonstrates the difference between the smoothed and original CHM, most pits have been removed. In detail (Figure B.6), the removal of pits in the canopies is more visible, and how the overall tree structure is kept intact.

Figure B.7c illustrates the differences in shading between the smoothed and original CHM. Values of 0.9 indicate areas where shading is present in the original CHM but absent in the smoothed version, while values of -0.9 indicate the opposite. Smoothing primarily reduces the shade size by removing peak values via the median filter, which slightly softens the shape, reducing the stripy effect slightly. In some cases, smoothing adds a thin line of new shadow as the area expands.

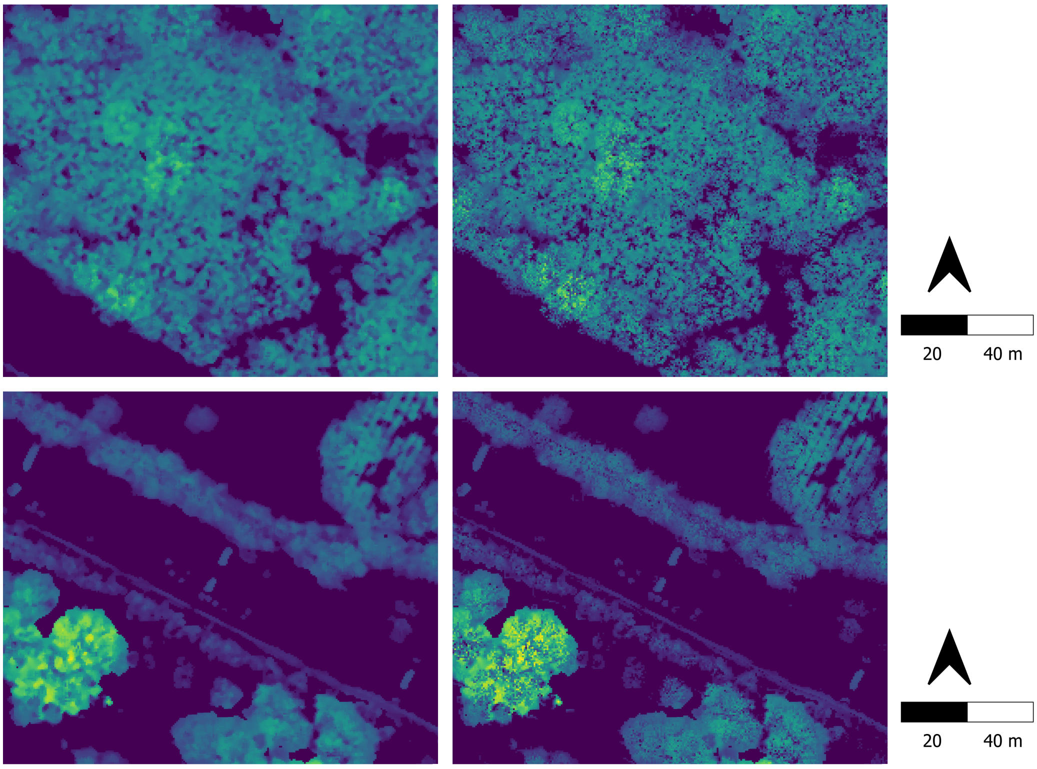
In conclusion, smoothing helps reduce pits within vegetation. These pits are mainly relevant for individual tree segmentation, a focus outside our scope, and smoothing does not speed up shade calculation. Thus, we have chosen to keep the CHM unsmoothed in our datasets.




(a) Smoothed

(b) Original

**Figure B.5:** The smoothed and the original CHM.




(a) Smoothed (b) Original

**Figure B.6:** The smoothed and the original CHM in detail.

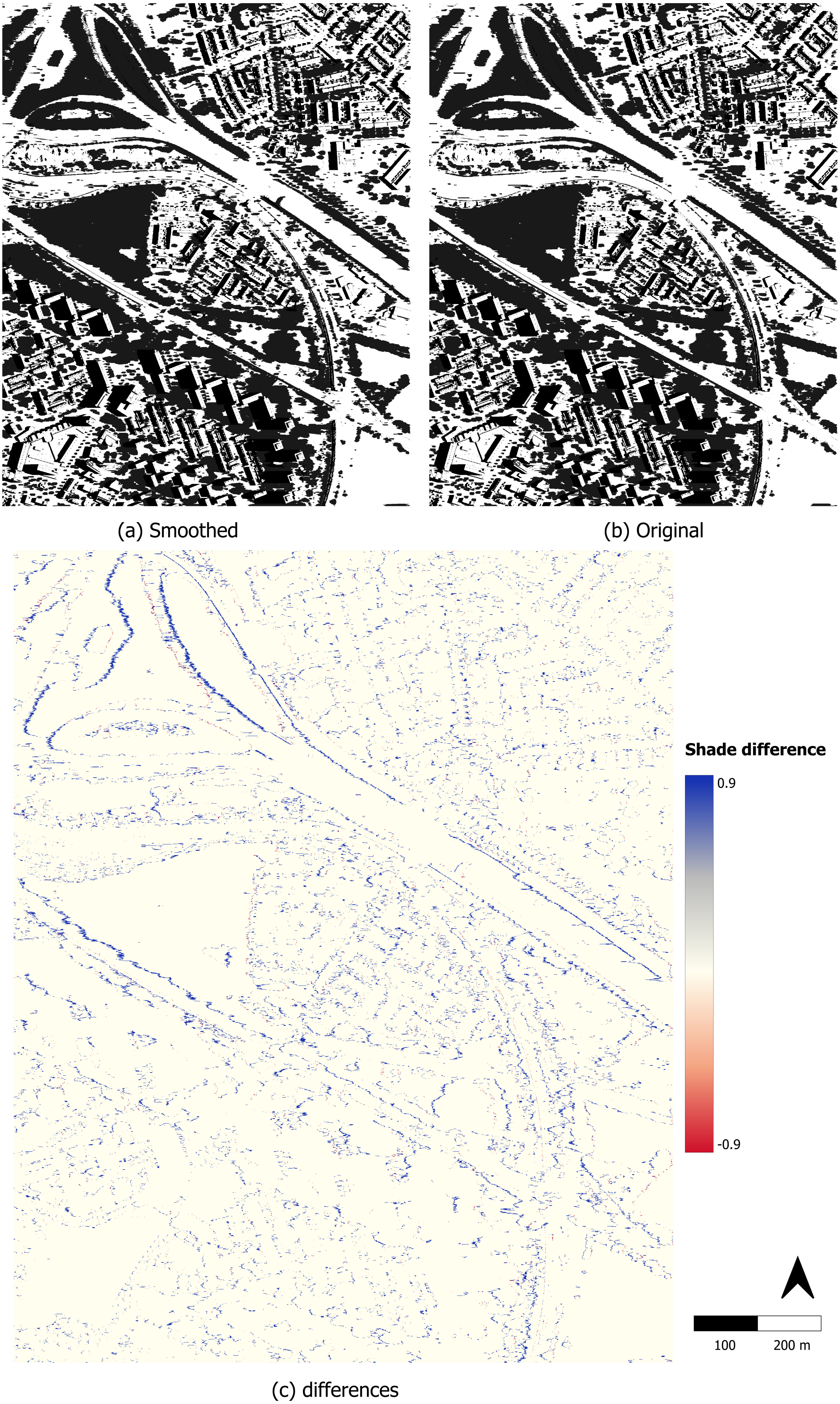


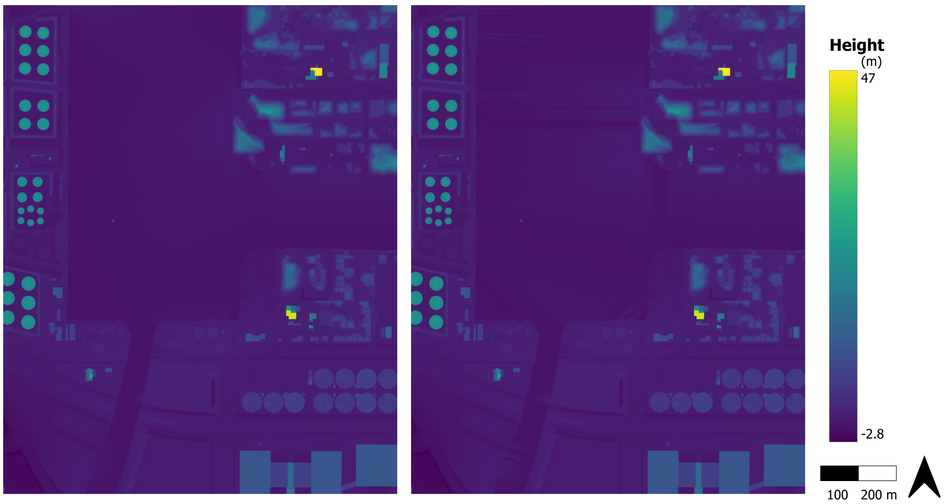

**Figure B.7:** The output shade differences between the smoothed and the original CHM in detail.

### B.2.3. Linear interpolation

Laplace interpolation can extend processing times, especially when large areas of ”no data” exist in the Triangulated Irregular Network (TIN). In our observations, interpolation in subtiles with large data areas can take up to 54 minutes, which is longer than the typical 3-5 minute range. A possible explanation might be the increased complexity involved in computing the interpolation, as numerous triangles surrounding the no data area must be evaluated to derive the interpolated value.

As a solution, we tried linear interpolation, which utilizes the triangle that contains the target location for interpolation. By weighting the vertices of this triangle using barycentric coordinates, the height at the desired point can be estimated (Ledoux et al., 2023). The results of this method, illustrated in Figure B.8, show height differences ranging between -1.2 and 1.2 meters (Figure B.8). Notably, the most significant height differences occur within the water surface, which corresponds to the original no data area. Since our shade analysis does not involve water surfaces, these variations are deemed acceptable. Additionally, implementing linear interpolation reduces the calculation time for the Digital Terrain Model (DTM) from 54 minutes to approximately 9 minutes. When processing is conducted in parallel, the few tiles that require longer interpolation times have minimal impact on overall efficiency, leading us to recommend against using this method when parallel processing.

If the speed-up method is implemented, the input cropped DTM undergoes an evaluation for connected regions of no data values. If a region exist with dimensions of 800x400m or 400x800m, linear interpolation is used.




(a) Laplace interpolation (b) Linear interpolation

**Figure B.8:** DSM created with laplace and linear interpolation.

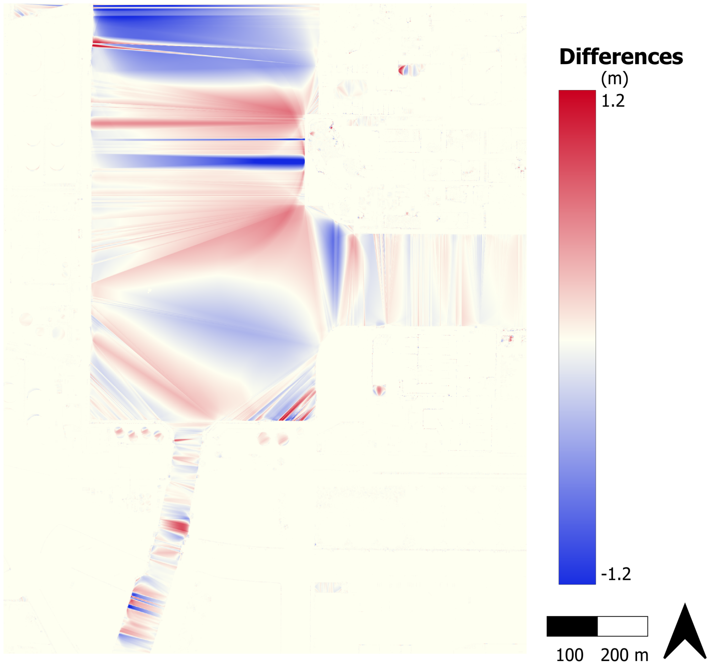



**Figure B.9:** Height differences between the DSM created with laplace and linear interpolation

## B.2.4. Reusing shade maps

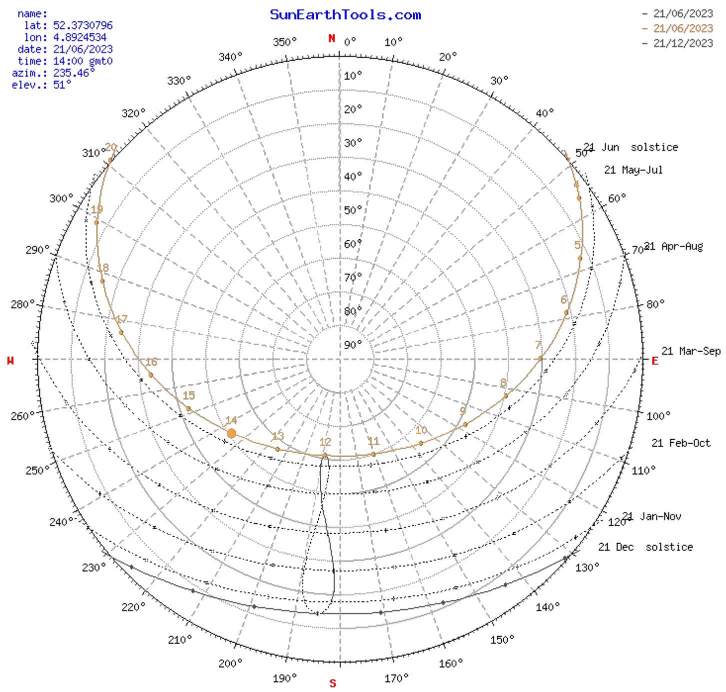
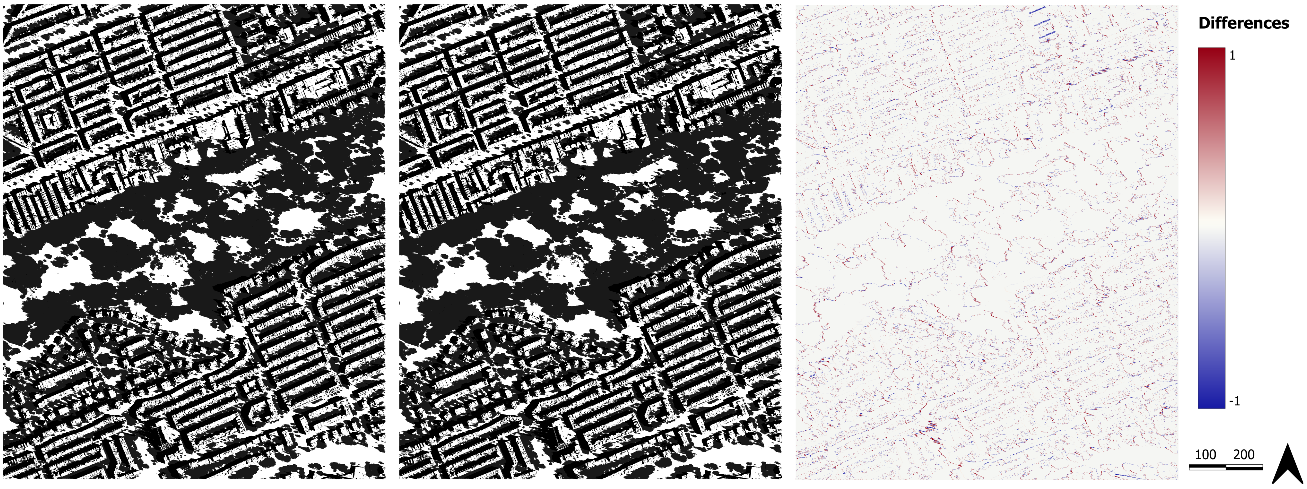



**Figure B.10:** The solar path

In our shade mapping, we approximate the shade at dates prior to the summer solstice using the calculated shades from dates following the solstices. However, the sun's position differs slightly before and after the solstice. To clarify this, we utilized SunEarthTools to analyze the sun position over Amsterdam for 2023 (Figure B.10). The analemma (the bowling pin shape below) represents the sun's position at noon for each day of the year. Although the elevation angles remain relatively consistent, there can be a maximum variance of 5 degrees in azimuth, resulting in slightly different shading angles.

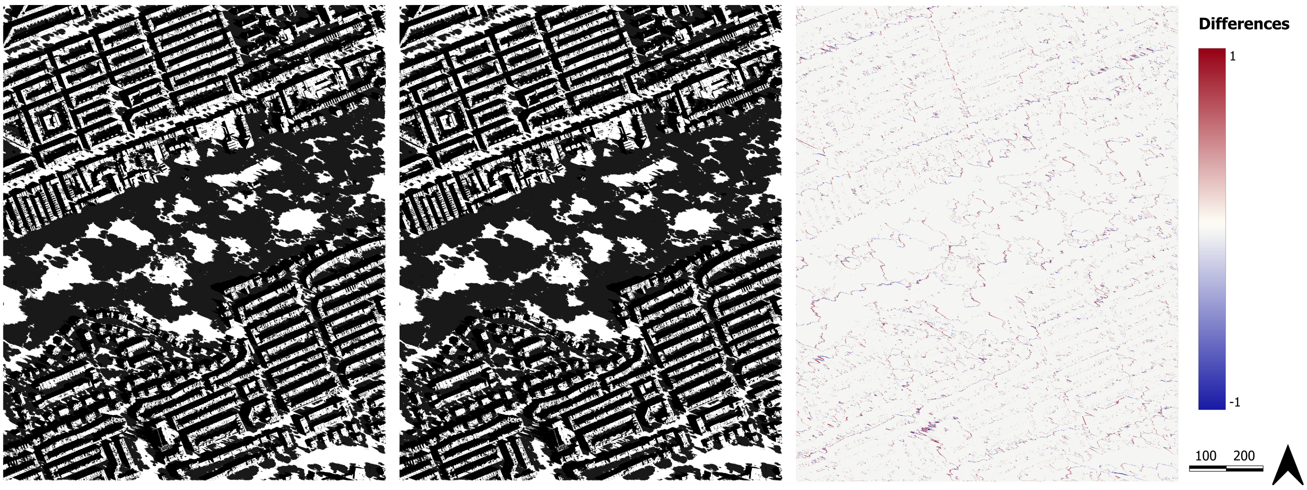
In the Figures B.11, B.12 and B.13, illustrate the shades for mirror dates in May-July, April-August, and March-September, respectively. Positive values indicate shades that only exist on the post-solstice dates, while negative values reflect those on pre-solstice dates. We observed slight shadow discrepancies at the edges of shade-casting objects, averaging about 1-2 pixels (equivalent to 0.5-1 meter). Notably, this difference increases for the two most distanced months.

We believe that the benefits of reduced computation and storage requirements outweigh the slight inaccuracies in this technique. Therefore, we have chosen to implement this approach.



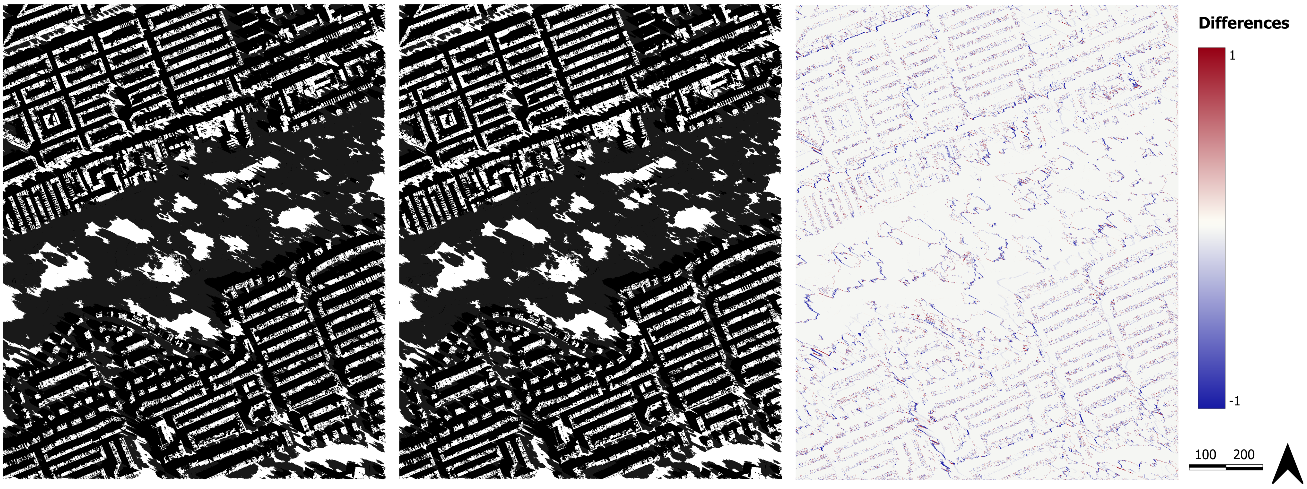

(a) 5-24 10 am (b) 7-19 10 am (c) Differences

**Figure B.11:** The calculated shade maps for May 24th (a), July 19th (b), and their shade difference (c).




(a) 4-26 10 am (b) 8-16 10 am (c) Differences

**Figure B.12:** The calculated shade maps for April 26th (a), August 16th (b), and their shade difference (c).




(a) 3-29 10 am (b) 9-13 10 am (c) Differences

**Figure B.13:** The calculated shade maps for March 29th (a), September 13th (b), and their shade difference (c).

## B.3. Tables

| Year | First Day Above 20°C | Last Day Above 20°C |
|---|---|---|
| 2010 | 2010-04-25 | 2010-10-04 |
| 2011 | 2011-04-02 | 2011-10-03 |
| 2012 | 2012-04-30 | 2012-10-22 |
| 2013 | 2013-04-14 | 2013-10-22 |
| 2014 | 2014-03-20 | 2014-10-19 |
| 2015 | 2015-04-15 | 2015-09-12 |
| 2016 | 2016-04-03 | 2016-09-28 |
| 2017 | 2017-03-30 | 2017-10-19 |
| 2018 | 2018-04-07 | 2018-10-17 |
| 2019 | 2019-04-07 | 2019-10-13 |
| 2020 | 2020-04-06 | 2020-10-21 |
| 2021 | 2021-03-30 | 2021-09-27 |
| 2022 | 2022-04-12 | 2022-10-30 |
| 2023 | 2023-05-04 | 2023-10-13 |

**Table B.1:** First and Last Days Above 20°C at the Bilt by Year

# B.4. Recommendation results for cool space

## B.4.1. Recommendation for June


**Figure B.14:** Recommendation result of June, shade (time) weight: 0.15, shade (area ratio) weight: 0.15


**Figure B.15:** Recommendation result of June, shade (time) weight: 0.3, shade (area ratio) weight: 0.0


**Figure B.16:** Recommendation result of June, shade (time) weight: 0.0, shade (area ratio) weight: 0.3

### B.4.2. Recommendation for July

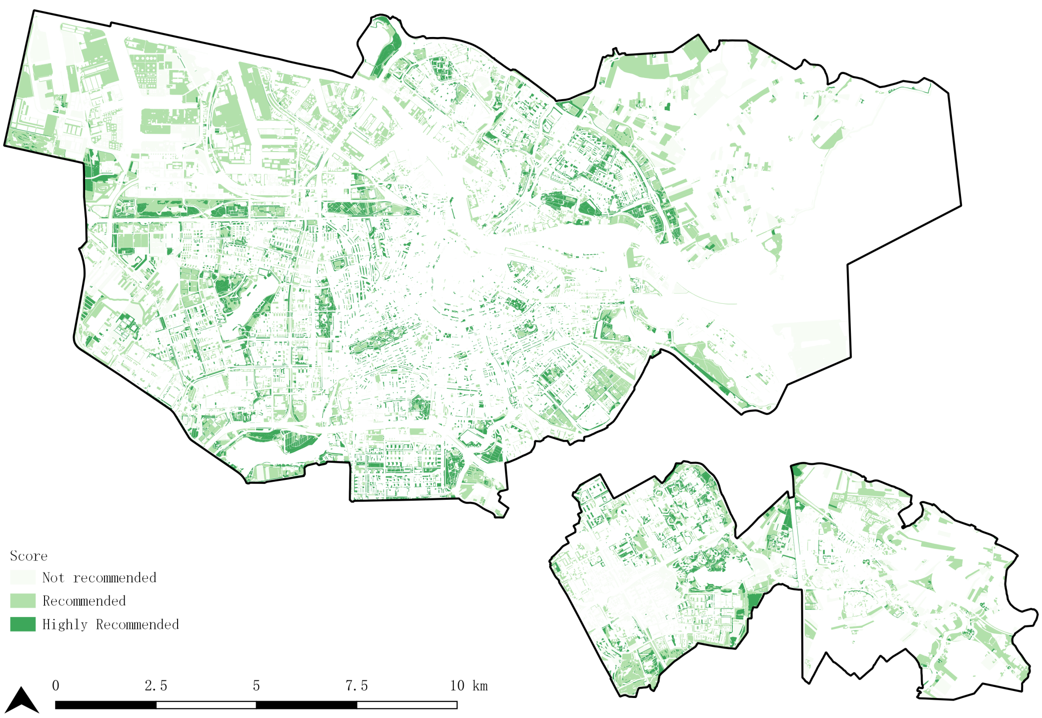



**Figure B.17:** Recommendation result of July, shade (time) weight: 0.15, shade (area ratio) weight: 0.15


**Figure B.18:** Recommendation result of July, shade (time) weight: 0.3, shade (area ratio) weight: 0.0


**Figure B.19:** Recommendation result of July, shade (time) weight: 0.0, shade (area ratio) weight: 0.3

### B.4.3. Recommendation for August


**Figure B.20:** Recommendation result of August, shade (time) weight: 0.15, shade (area ratio) weight: 0.15


**Figure B.21:** Recommendation result of August, shade (time) weight: 0.3, shade (area ratio) weight: 0.0


**Figure B.22:** Recommendation result of August, shade (time) weight: 0.0, shade (area ratio) weight: 0.3

### B.4.4. Recommendation for September


**Figure B.23:** Recommendation result of September, shade (time) weight: 0.15, shade (area ratio) weight: 0.15


**Figure B.24:** Recommendation result of September, shade (time) weight: 0.3, shade (area ratio) weight: 0.0


**Figure B.25:** Recommendation result of September, shade (time) weight: 0.0, shade (area ratio) weight: 0.3

## B.5. Details of cool space recommendation results

In order to reserve non-green public space, the land use data did not filter out non-green land use types, this decision resulted in both good and limited recommendation results. On the positive side, which is shown in figure B.26, some non-green public spaces are correctly identified and evaluated. For example, the bottom right sub-figure shows the public spaces of a casino in downtown Amsterdam. It clearly reflects the current lack of shade in some of the public spaces in this area, which makes them not recommended cool spaces. For the green space, the bottom-left sub-figure shows a recommendation result of a park, and it also clearly reflects which area of the park lacks shade.

**Figure B.26:** Examples of good cool space process results. Top left and bottom left: evaluation results of green space. Top right and bottom right: evaluation results of non-green space.

On the other hand, due to the lack of detailed categorization of land use data, some public spaces that are not suitable for recreation are also included. For example in figure B.27, the top-left and bottom-left sub-figures clearly show the incorrect recommendation results of parking lots. This is because, in the land use data, these parking lot areas are mixed with the surrounding areas which means they are not classified independently but as part of residential area. Besides, due to the lack of building geometries in certain parts of Amsterdam, the cool space results did not filter out building areas, resulting in incorrect recommendations.

A more complex issue is sometimes it can be hard to define public and private areas by land use classifications such as "bebouwd gebied (built environment)". The top-right sub-figure of figure B.27 shows examples of these areas, which are mostly downtown residential areas in Amsterdam. By the attribute name, they seem to be public but in fact they are private neighborhood areas. Although this mistake can be easily solved by simply filtering out all "bebouwd gebied" areas, it is still worthy considering how the precision of the land use classification can affect the final cool space recommendation results.

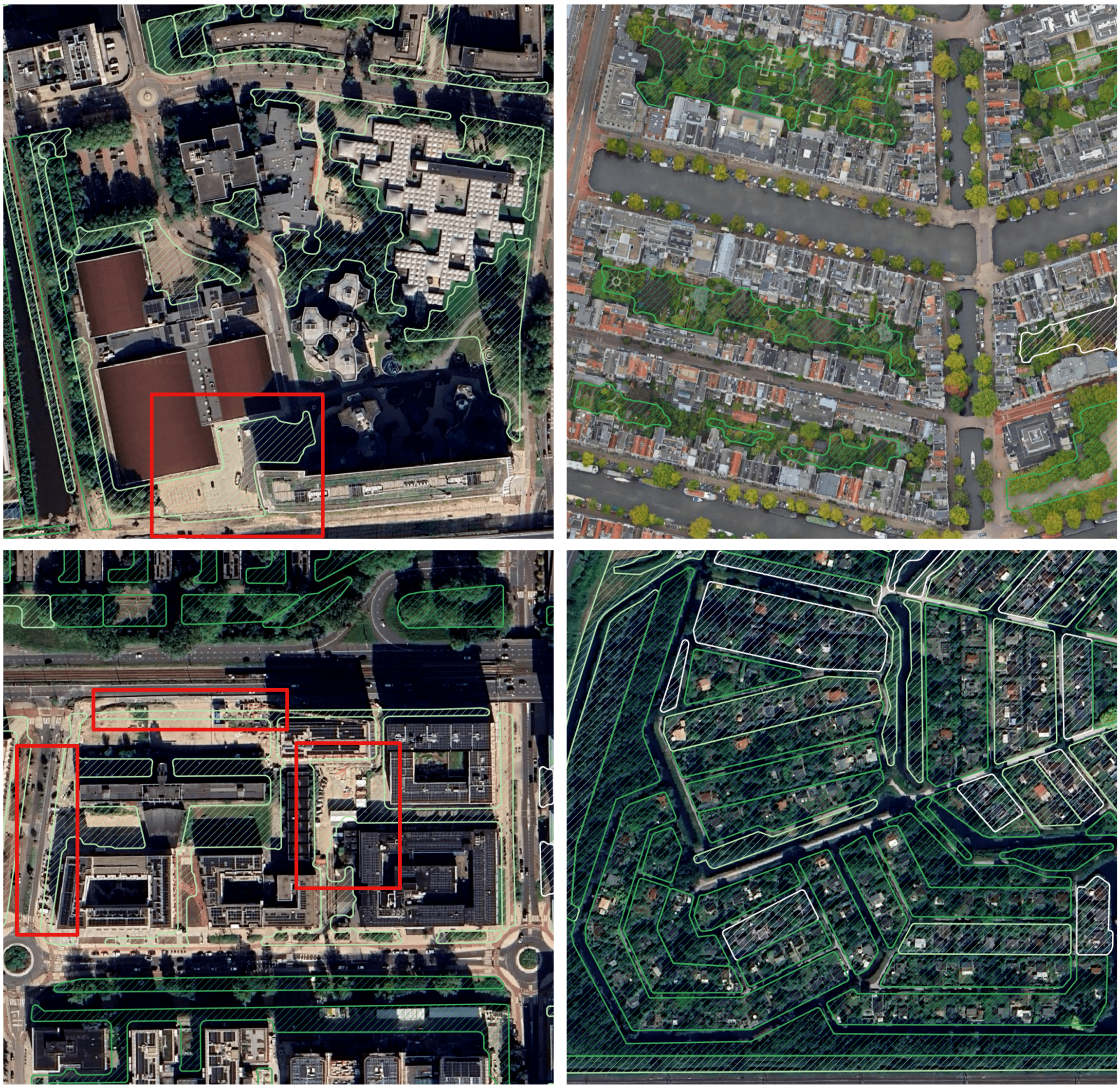


**Figure B.27:** Examples of bag cool space process results. Top left and bottom left: incorrect recommendation of parking lots. Top right: incorrect recommendation of private space. Bottom right: incorrect recommendation of areas containing buildings.