MSc thesis in Geomatics

Thermally conscious urban mobility: operationalizing urban microclimate data into a pedestrian routing tool

Anne den Hartog - 5096243

June 2026

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

The work in this thesis was carried out in the:

3D geoinformation group
Delft University of Technology

Supervisors: Clara Garcia-Sanchez
Daniela Maiullari
Lukas Beuster

Co-reader: Akshay Patil

Abstract

This thesis investigated how SOLWEIG-derived urban microclimate data can be transformed and integrated into a fully open-source, scalable pedestrian routing tool that supports thermally conscious mobility decisions. Scalability is treated as a central design requirement because such a tool can only contribute to climate-adaptive urban mobility if it can be reproduced and extended to different areas without being limited by manual processing or computational bottlenecks.

The modular workflow prepares SOLWEIG input data, runs SOLWEIG_GPU to generate hourly UTCI outputs, samples these outputs onto an OpenStreetMap-derived pedestrian network as edge-level thermal-comfort attributes, and uses these in a modified, state-aware weighted Dijkstra algorithm with user-defined preferences. Interaction with the route planner is implemented using a locally deployable web interface. Evaluation with stratified origin-destination pairs shows that thermally conscious routes can reduce modelled UTCI exposure compared with the shortest path, particularly during hotter hours, while requiring only limited additional walking distance. The development process further shows that scalability depends not only on the routing algorithm, but also on data structures, file formats, network representation, and modular software design.

Acknowledgements

I would first like to thank my supervisors for their guidance, feedback, enthusiasm, and encouragement throughout this thesis. Their insights helped me shape this work into the strongest version of itself, while their understanding and support also helped me navigate the challenges beyond the academic aspects of the thesis. Sincerely, thank you!

I would also like to thank Julia and Michel for keeping me accountable (and present in the geolab) throughout the process, without you I doubt I would have been able to finish on time.

Lastly, I'd like to thank my friends and family for their encouragement and patience.

Contents

List of figures

List of tables

List of Abbreviations

API	Application Programming Interface
CHM	Canopy Height Model
DEM	Digital Elevation Model
DSM	Digital Surface Model
DSR	Design Science Research
DTM	Digital Terrain Model
IPCC	Intergovernmental Panel on Climate Change
OD	Origin-Destination
OSM	OpenStreetMap
OTC	Outdoor Thermal Comfort
PET	Physiological Equivalent Temperature
PyPI	Python Package Index
SOLWEIG	Solar and LongWave Environmental Irradiance Geometry
UMC	Urban Micro Climate
UMEP	Urban Multi-scale Environmental Predictor
UTCI	Universal Thermal Climate Index

1. Introduction

Climate change is increasing the frequency and intensity of extreme weather events, including heatwaves, which pose growing risks to human health. Urban areas are especially vulnerable to the effects of heat due to their materials and phenomena such as the urban heat island effect[32]. As highlighted by the Intergovernmental Panel on Climate Change (IPCC) in 2022, the high concentration of people in cities means that climate impacts affect large populations simultaneously, but also that interventions in urban areas can deliver substantial benefits for the same reason. Improving the adaptive capacity of urban communities can therefore play a key role in managing the effects of climate change. [13]

In everyday urban life, heat-related climate impacts can directly affect thermal comfort during outdoor activities, such as walking. With thermal comfort being defined as *"… environmental conditions where an individual feels no thermal stress and exhibits no sign of thermal strain. In these circumstances, the net heat gain/loss of the body will be close to zero."* [29]. This state occurs from the interaction between the human body and its surrounding environment. So when the environment changes in temperature, the body experiences thermal stress. In outdoor settings, this concept is commonly referred to as Outdoor Thermal Comfort (OTC).

Poor outdoor thermal comfort can discourage walking and increase heat-related health issues, especially among vulnerable populations like older people, children, and individuals with pre-existing health conditions [10]. This is why OTC is not only related to environmental quality but also to public health and social equity [9]. Outdoor thermal comfort is influenced by a wide range of parameters which can be grouped in two categories: environment-based (e.g., weather and climate conditions) and human-based (e.g., physiological characteristics) [1] (Section 3).

Environment-based factors shape the outdoor thermal conditions that pedestrians are exposed to, while human-based factors influence how individuals perceive those conditions. In cities, the environmental factors can vary substantially over short distances and time scales due to building geometry, surface materials, vegetation, and wind patterns. For example, even in a single street, the thermal conditions can differ wildly between a shaded area and a non-shaded area or between times of day. This spatial and temporal variability makes OTC particularly challenging to assess using measurements alone, and highlights the opportunity for urban microclimate modelling approaches to analyze OTC across space and time [28].

Urban Micro Climate (UMC) modelling uses various inputs to model environmental conditions in a small spatial and temporal resolution. A model that has solvers relevant for OTC is the open-source UMEP developed by Lindberg et. al (2017). In this thesis, the SOLWEIG solver is used to estimate the mean radiant temperature [17]. Which, in combination with weather information form ERA5, gives spatially explicit information on thermal stress through a Universal Thermal Climate Index (UTCI) map. This index, expresses the combined effect of air temperature, humidity, wind speed, and radiation on the human body as an equivalent temperature. It is therefore useful for this research because it translates complex microclimatic conditions into a single indicator that can be used to compare the experienced heat exposure.

Tools like UMEP have primarily been used for research and long-term urban planning/ design rather than everyday decision-making for individuals [22, 23]. To support this everyday decision-making, this thesis focuses on integrating environmental data generated by UMEP

into an open-source pedestrian routing tool. With the ultimate goal of improving the adaptive capacity of communities by supporting climate-aware mobility decisions.

Previous work on thermal comfort-oriented routing is characterized by differing methodologies, both in input data collection and routing algorithm design (Section 4). These differences are partly caused by the data (or lack thereof) that was available for researchers to collect or use. Additionally, much of the existing research focuses primarily on demonstrating that comfortable routes can offer measurable benefits [10, 18, 39]. Which provides important proof of concept insights. However, there has been less emphasis on the reproducibility, scalability, and openness of the underlying systems. As a result, most of the proposed approaches are either difficult to scale to new cities or have not been implemented in user-facing applications at all.

This thesis addresses these limitations by focusing on the development of a fully open-source tool for thermally comfortable pedestrian routing. First, the challenge of environmental data availability is addressed through the use of UMEP / SOLWEIG to model the UTCI. Secondly, the resulting UTCI data are used by a routing algorithm that generates thermally conscious routes for users. Throughout this process, the thesis documents and evaluates how complex urban microclimate data, like UTCI maps, can be used by a scalable pedestrian routing tool.

2. Research questions and scope

2.1. Problem statement

Outdoor thermal comfort varies substantially across space and time, making models such as SOLWEIG useful for estimating fine-scale thermal conditions. Pedestrian routing provides a practical use case for operationalizing SOLWEIG outputs by translating modelled thermal conditions into thermally conscious routes. Existing research on thermally comfortable routing differs widely in input data and algorithmic design, with a focus on showing that thermally conscious routing can provide measurable benefit to OTC along the route. However, there is a lack of attention to the reproducibility, scalability, and openness of the underlying systems. This thesis addresses these gaps by developing an open-source workflow that connects SOLWEIG-based UTCI modelling with a pedestrian routing algorithm capable of generating thermally conscious routes. The workflow is evaluated not only in terms of whether the generated routes reduce thermal exposure, but also in terms of how microclimate outputs can be stored, processed, and integrated into a scalable pedestrian routing tool.

2.2. Research question

All this leads to the following research questions:

- **RQ:** How can urban microclimate data generated by UMEP's SOLWEIG be transformed and integrated into a fully open-source, scalable pedestrian routing tool to support thermally conscious mobility decisions at the individual level?
  - **RQ1:** What design strategies and data structures enable the effective use of UMEP outputs in a scalable pedestrian route planning application?
  - **RQ2:** How can established weighted-cost routing algorithms incorporate SOLWEIG outputs and be implemented and adapted to generate thermally conscious pedestrian routes?
  - **RQ3:** What does scalability mean for a pedestrian routing tool particularly regarding its potential for spatial and functional extension?

2.3. Scope

This thesis follows a design-based approach, in which the goal is to develop and evaluate a workflow for incorporating UMEP/SOLWEIG outputs into a pedestrian routing tool. The focus is therefore not only on the resulting routes, but also on the design strategies needed to transform microclimate model outputs into usable network attributes for a scalable pedestrian routing application. Due to the computational and data-processing complexity of high-resolution urban microclimate modelling, the spatial scope is limited to a case-study area in Rotterdam. This limitation makes it possible to investigate the workflow in depth, with the scalability of the underlying data structures, routing logic, and software architecture remaining an important design principle throughout the implementation.

The microclimate component is limited to the use of SOLWEIG-derived outputs for estimating thermal stress through UTCI. The temporal scope is further restricted to high-heat conditions (specifically July 19th 2022, which had an average temperature of 29,2°C [16]). This focus is motivated by the association between heat stress and negative health outcomes, making heat exposure the most relevant thermal-health concern to investigate for this application. The routing component is limited to pedestrians.

2.4. Approach

First this thesis will provide the necessary background information on concepts related to OTC and Dijkstra's algorithm (Section 3). This will explain the basics such that it is clear how this thesis adjusted or used the original concepts. Then the next section will give context on routing in urban areas, focussing on environmental weights and how people navigate through a city to begin with (Section 4). From this follow the methodology, which explains both how the thermally conscious route planner was built and how insights were gathered from the development process (Section 5). Where relevant, these methodological steps are then described in greater technical detail to support the reproducibility of the workflow (Section 6).

After the tool development, the new generalized insights will be discussed as well as an analysis of the thermally conscious routes in comparison to the shortest route (Section 7). Lastly, the answers to the research questions will be given as well as the specific contributions and limitations of this research. Followed by the recommendations for future work(Section 8).

3. Background

This section introduces the theoretical background needed to understand the main concepts used throughout the thesis.

3.1. Health outcomes & heat

In the same weather conditions, the form and material characteristics of the urban environment can amplify thermal discomfort for people living in cities when compared to rural areas [28]. Especially heat stress, which becomes more relevant as extreme weather events become more likely with climate change [13], is an area of concern. Extreme (or even moderate) heat goes hand in hand with an increase in mortality [28] and negative health outcomes [1, 41]. To assess human thermal stress in outdoor urban environments, the following sections first introduce the broader concept of outdoor thermal comfort (OTC) and then discuss the Universal Thermal Climate Index (UTCI) as a metric for quantifying thermal stress.

3.1.1. OTC

A state of thermal comfort occurs when the body is in thermal balance with its environment and does not need to actively compensate for heat gain or heat loss. In outdoor settings, a wide range of factors influence thermal comfort, which can broadly be grouped into two categories:

- Environment-based: weather and climate conditions (e.g., temperature, wind, humidity), but also the secondary physical characteristics (e.g., morphology, geometry, material properties).
- Human-based: physiological characteristics (e.g., gender, age, clothing), but also psychological and cultural factors can play a role. [1]

The environmental factors create the local microclimate conditions that pedestrians find themselves in. While the human factors help explain how people subjectively perceive and adapt to these environmental conditions. In general these parameters help explain why thermal comfort preferences vary between population groups and climates.

3.1.2. UTCI

The interaction between outdoor thermal conditions and the human body can be understood through the human energy balance, *"… which is a complete description of the biophysical processes that underpin the thermal state of the body"*[28]. Building on this principle, the UTCI combines a thermophysiological model with a clothing model and a predefined reference condition of a person (e.g. walking at 4 km/h). This results in a one-dimensional temperature-scale index that represents the heat stress experienced by the human body [6].

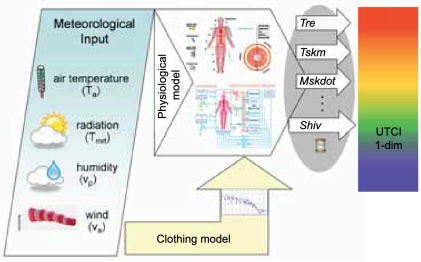

Figure 3.1: Concept of the Universal Thermal Climate Index (UTCI). Reproduced from Błażejczyk et al. (2013) [6]

3.1.3. Urban Microclimate Modelling

Urban Microclimate (UMC) modelling tools operationalise these concepts by simulating the environmental variables that govern the human energy balance in outdoor urban environments. In this thesis, the UMEP is used to model these conditions, because it is an open-source tool [17]. From UMEP, the SOLWEIG (SOlar and LongWave Environmental Irradiance Geometry model) model is used to estimate mean radiant temperature by modelling spatial variations in shortwave and longwave radiation fluxes in complex urban settings [17]. Which is used as input for the UTCI calculation together with ERA5 weather data.

3.2. Dijkstra's least cost algorithm

Dijkstra's algorithm is a well-known greedy algorithm for finding shortest paths in a weighted network with non-negative edge weights [8]. It was introduced by Dijkstra in 1959 and remains a common basis for routing applications.

The algorithm works by maintaining a tentative distance for each node. At the start, the distance to the source node is set to zero, while all other nodes are assigned an infinite distance because no path to them has yet been found. The algorithm then repeatedly selects the unvisited node with the lowest tentative distance, which is kept at the top of the priority queue. This greedy choice is safe because, with non-negative edge weights, no later path can reduce the distance to that node once it has been selected. From this node, the algorithm evaluates all outgoing edges and updates the tentative distances of neighbouring nodes when a shorter path is found. The process continues until all reachable nodes have been visited, or until the destination node has been reached. The full algorithm in pseudo-code is presented in Listing 1.

```
function Dijkstra(Graph, source, target):

    dist[source] ← 0
    prev ← empty dict
    processed ← empty set

    Q ← priority queue
    Q.push(0, source)

    WHILE Q IS NOT empty:

        current_cost, u ← Q.pop()

        IF u is in processed:
            continue

        processed.add(u)

        IF u = target:
            break

        FOR EACH neighbour v of u:
            edge_cost ← weight(u, v)
            new_cost ← current_cost + edge_cost

            IF v NOT IN dist OR new_cost < dist[v]:
                dist[v] ← new_cost
                prev[v] ← u
                Q.push(new_cost, v)

    RETURN (dist, path)
```

Listing 1: Dijkstra's algorithm in pseudocode

4. Related Work

This chapter situates the thesis within the broader research by focusing on two topics. First, it discusses research on unconscious movement patterns in cities and the influence of the urban environment on these route choices. Secondly, papers are discussed that use differing 'environmental factors' to support citizens in making these decisions consciously through environmental informed routing. There are three main elements to these tools: the input environmental data, the pedestrian network, and the routing algorithm.

4.1. Mobility patterns

Human mobility patterns are recurring regularities in movement on a population scale. The fact that these routes can be identified indicates that some routes are generally preferred over others [36]. These patterns in mobility follow from a multidisciplinary set of forces, including *"socioeconomic conditions, technological advancements, policy interventions, and environmental changes"* [9]. Increased availability of GPS data has helped develop this area of research by providing precise route information. Analysing these common routes, and their reasons for deviating from the shortest path, can support more informed routing decisions at both the individual and collective levels [31]. For example, by routing both car traffic and pedestrians in such a way that pedestrian exposure to air pollution is minimized [3].

4.2. Unconscious route choice

Routing choices in urban areas are not solely dependent on minimizing the distance in meters; factors like facade complexity, nearness to greenery, or other urban features can also play a role [31]. These features can shape how long or demanding a route is perceived to be, creating a difference between measured and perceived distance. This is described by the concept *cognitive distance*, the distance humans *think* a route will be, illustrated by Figure 4.1 [20]. Approaching this concept practically, Salazar Miranda et al. (2021) used GPS data from mobile devices to investigate what makes people deviate from the shortest path. They found that urban greenery is the most common feature on these deviating routes [36]. Many of these routing choices are not done consciously [10]. However, studies show that thermal conditions also influence route choice. With Basu et al. (2024) showing that, when given the option, pedestrians tend to pick routes that avoid heat stress [4].

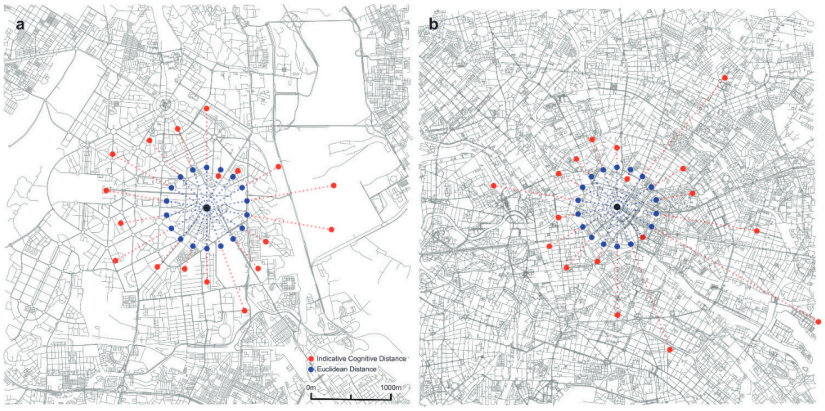

Figure 4.1: *"Plots of indicative cognitive distance (red points) and Euclidean (blue points) distances from a single origin point, in Delhi (left) and Berlin (right)."* (Manley et al. 2021) [20].

4.3. Environmentally informed routing tools

4.3.1. Input data

Choosing a route that reduces exposure to environmental factors, like heat stress or pollution, can provide statistically different paths to the shortest path [2, 10, 18, 25, 35, 39]. However, which environmental input data is used and in what form differs largely between papers. With some papers using static maps containing the environmental variable(s) [2, 25, 27]. Rußig & Bruns (2017) used data collected by a thermal scanner flight adjusted with hourly data from a local weather station [35]. Some of the most extensive data collection was done by Foshag et al. (2024), who used a collection of multidisciplinary methods to investigate heat-sensitive routing with a special focus on vulnerable groups. By using a combination of surveys, interviews, shade maps, and modelling using ENVI-net they built a route planner that avoids heat stress [10]. This gives a very detailed implementation for one area that can help policy makers, but is cost-intensive to replicate on a large scale. Alternatively, Novack et al. (2018) built their pleasant route planner based entirely on open street map (OSM) data. This supports openness and reproducibility, but they only base their routes on noisiness, closeness to greenery, and social places [27].

4.3.2. Routing algorithms

In existing literature, pedestrian networks are typically generated using OpenStreetMap (OSM) data, creating a topologically connected network of all paths in a city. Routing is commonly performed using weighted shortest-path algorithms such as Dijkstra [2, 35, 39] or A* [25], although in some studies the specific algorithm is not explicitly stated [10, 27]. Before application of the routing algorithm weights need to be assigned to the edges in the network. The methods used to compute these weights differ substantially between studies.

The most straightforward approach is to assign each network edge a static cost based on a weighted combination of relevant factors, such as thermal conditions, distance or greenery [10, 25, 27]. In this approach, all edge weights are calculated before the routing algorithm is applied and remain fixed during route generation. This gives a formula for the weight of an edge W with f_i being the factors you wish to consider (e.g., distance, shade, greenery) and w_i being the weights assigned to each of these factors.

$$W = \sum_{\{i=1\}}^{\{n\}} w_i f_i$$

Several studies extend static edge weighting to better account for the dynamic character of thermal comfort during walking. A pedestrian's thermal experience is influenced not only by the current edge conditions, but also by their physiological state and recent exposure history. As walking continues, physical exertion can increase the demand for cooler edges, as the body heats up [18, 35, 39]. Similarly, whole-trip thermal comfort research shows that current thermal sensation is influenced by recent previous thermal experiences, so the perceived cost of an edge may depend on the thermal conditions encountered immediately before it [42]. This suggests that routing approaches should allow edge weights to dynamically change during route generation, rather than being fixed in advance.

To achieve this Wen et al. (2025) adjusted the generated route using a sigmoid function, deduced from a pedestrian movement dataset, to find an optimal balance between actual distance and sun exposure [39]. While Rußig & Bruns (2017) use a time-dependent multi-layered raster as the input to account for this effect. Aliyev & Nanni (2025-6) applied a different technique to find an optimum between distance and exposure, though they wanted to avoid

pollution. They generated a set of routes with different weights, to use a statistical method to find the optimum between distance and exposure for every trip [2]. This was later extended to dynamically adjust the weights as pollution moves trough the city [3]. Although demonstrated for pollution avoidance, this approach allows weights to be adjusted automatically and to vary between individual trips. These approaches in routing algorithm are summarized in Table 1. Descriptions

- **Static** the previously visited edges do not have an impact on the rest of the route.
- **Dynamic** the previously visited edges can change on the weight of the next edge.
- **Predefined** weights for factors (e.g. distance, shade) are pre-emptively assigned to edges by the researchers.
- **User defined** weights for factors (e.g. distance, shade) are defined by the user of the system.
- **Automatically derived** weights for the factors are automatically derived by an optimization problem for every trip.

	Edge weights	**Algorithm**
Mora-navarro et al. (2018)	Static, predefined	A*
Novack et. al (2018)	Static, defined by user	Unspecified
Foshag et al. (2024)	Static, predefined	Unspecified
Rußig & Bruns (2017)	Dynamic, predefined	Dijkstra
Wen et al. (2025)	Dynamic, predefined	Dijkstra
Aliyev & Nanni (2025-06)	Static, automatically derived	Dijkstra
Aliyev & Nanni (2025-10)	Dynamic, automatically derived	Dijkstra

Table 1: Algorithm descriptions papers

The differences between the algorithmic approaches even in recent papers show that this remains an open methodological problem rather than a settled solution.

5. Methodology

This section first outlines the research approach, contextualising the methodological choices made in this thesis. It then presents the study design in a series of larger methodological steps, followed by a description of the data sources and tools used. Finally, the individual software components that together form the final routing tool are described.

5.1. Research approach

This thesis adopts a Design Science Research (DSR) methodology in which knowledge is generated through the iterative design, implementation, and evaluation of an artefact [34, 40]. Here that artefact is an open-source pedestrian routing tool that integrates UMEP outputs to generate a thermally conscious route alternative to the shortest path. The final tool serves both as a result of the research and as a way to investigate strategies for integrating urban microclimate data into a pedestrian routing application. To ensure that the design process yields generalizable insights, systematic evaluation is conducted throughout all stages of development [33].

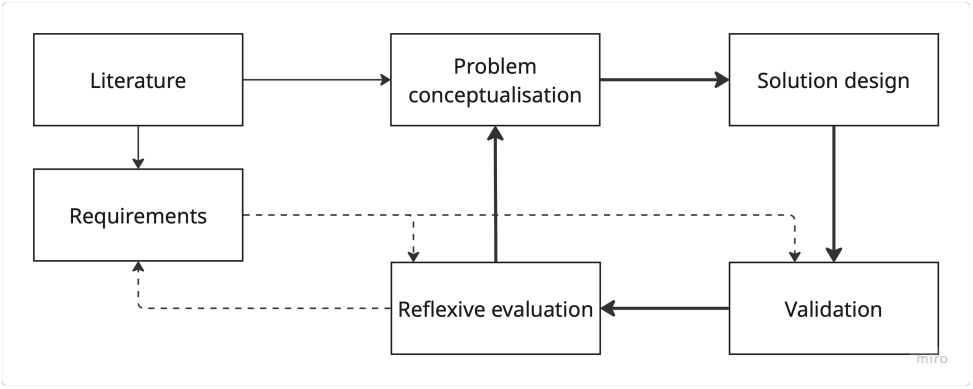

Figure 5.1: Overview methodological approach

Figure 5.1 shows how this iterative development is applied in this thesis. First literature is used to conceptualize the problem and identify the main design requirements. Then a solution is designed, which is validated according to the previously defined requirements. After which reflexive evaluation is used to readjust the problem conceptualization and requirements for the next design iteration.

This process reflects the practical orientation of design science research for software engineering. Where the contribution of the research stems from its ability to solve a problem [34]. This idea is encompassed in the concept of utility, as described by Hevner et al.: *"If the artifact does not solve the problem (search, implementability), it has no utility. If utility is not demonstrated (evaluation), then there is no basis upon which to accept the claims that it provides any contribution"* [12].

Utility is not only achieved through formal validation against predefined requirements, but also through insights generated by the iterative process with which the final tool is built. This aligns with the notion of emergent design, in which methodological choices are allowed to develop in response to findings that arise during the research process. [26]. This means that insights gained during the process can change the process itself and are expected to shape the direction of inquiry.

5.2. Study Design

Existing research has shown the possibility and usefulness of thermally informed routing [10, 18, 35, 39], but paid less attention to the technical foundations. To address this gap, this thesis considered the full workflow from microclimate modelling to route generation. This structure makes it possible to assess both the utility of the final route planner and the technical requirements of the individual components on which it depends.

5.2.1. Stages

1. A literature review was conducted to identify the functional requirements of the tool. This review informs the core components of the application, the role of SOLWEIG in the workflow, and the desired routing logic. It also establishes starting technical requirements with which the components are validated.
2. A testing area was chosen in Rotterdam for which the prototype was developed.
3. The route planner is developed iteratively with separate but connected components. This allows each component to be designed, analysed, and refined before being integrated into the complete system. Here the emergent nature of the research leaves room for new requirements and directions of inquiry as the implementation develops.
4. The resulting route planner is evaluated with regard to its utility and the requirements derived from the literature review. Utility is understood here as the practical, goal-oriented value of the tool: the extent to which it performs the task it was designed to perform. The evaluation therefore considers both whether the route planner achieves its intended routing function and whether it satisfies the technical requirements identified from the literature.
5. The development process is reflected upon to identify key challenges, trade-offs, and effective implementation choices. These reflections are used to derive broader insights and practical recommendations for future UMC-informed routing applications.

5.2.2. Study area

For the creation of the tool, a bounding box was chosen in a neighbourhood in Rotterdam with varying urban features, including green and blue corridors, parks, and different types of roads. These features are relevant because urban morphology and greenery can influence pedestrian-level thermal conditions. In particular, the presence of trees, open spaces, water, and different street geometries makes this area suitable for testing whether the routing tool can respond to the variation caused by these features in the modelled thermal comfort [22]. As displayed in Figure 5.2 the bounding box spans about 2.5 kilometres north and 2 kilometres east to west. This is to limit the calculation time for SOLWEIG and data download. An area of this size allows for in-depth testing without great demands on runtime.

The bounding box has the following coordinates ((left lower),(right upper)):

- **WSG84 (lat/long)**: ((51°52′46.75″,4°27′39.74″), (51°53′48.48″,4°28′59.42″))
- **Amersfoort / RD New**: ((91229,432752),(92777,434641))

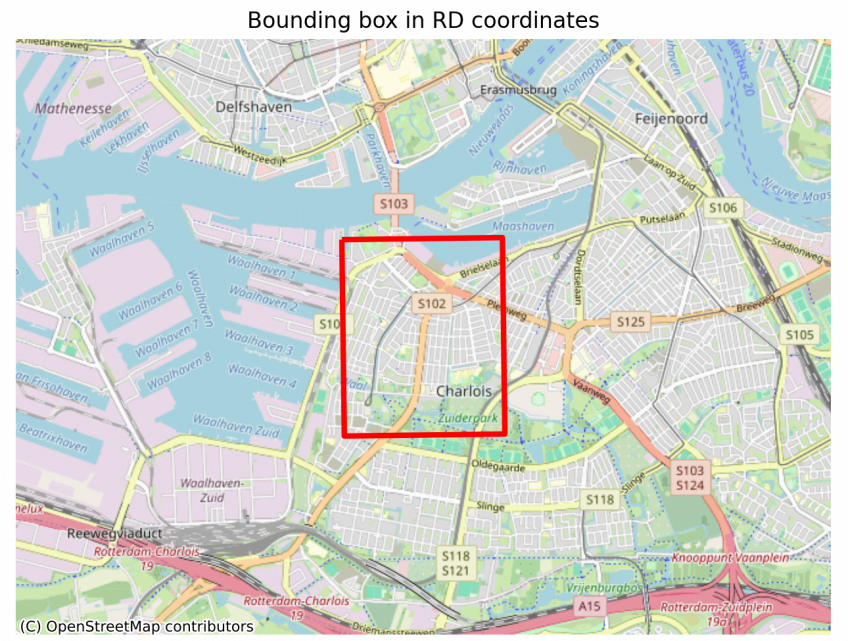

Figure 5.2: Red bounding box area for testing

5.2.3. Data

- **Open Street Map (OSM) [5]:** Base map used for the pedestrian network.
- **ERA5:** ERA5 is a global reanalysis dataset that provides consistent hourly estimates of past meteorological conditions. The simulations were forced with hourly ERA5 meteorological input data for 19 July 2022. The main climate variables used were air temperature, relative humidity, wind conditions, and radiation-related variables, which together define the hourly atmospheric conditions used in the UMEP/SOLWEIG simulations.

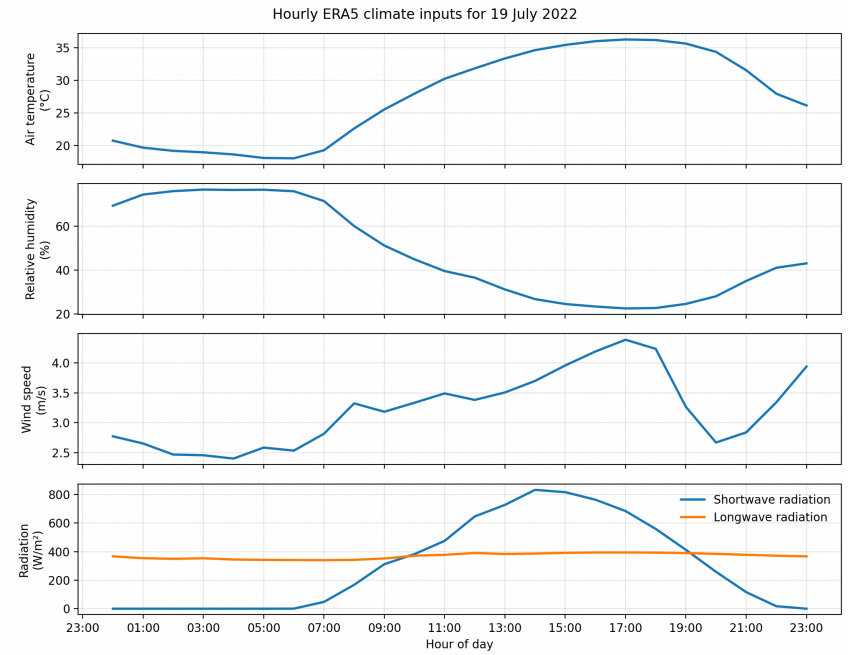

Figure 5.3: Hourly ERA5 climate inputs used for the UMEP/SOLWEIG simulations on 19 July 2022. Values represent the spatial mean of the ERA5 grid cells covering the study area. Radiation variables are shown as hourly mean fluxes, converted from accumulated energy values.

5.2.4. Tools

- **SOLWEIG [17]:** Urban microclimate model that estimates thermal comfort on a micro scale.
- **SOLWEIG_GPU [14]:** Implementation of SOLWEIG [17] on GPU that achieves speed-up in terms of computation time.
- **DelftBlue [7]:** high performance supercomputer TU Delft that was used to do the SOLWEIG_GPU runs.
- **SOLFD [23]:** Pipeline for generating all the input necessary to run SOLWEIG when given only a bounding box.
- **UMEP [17]:** Urban Multi-scale Environmental Predictor, open-source tool that houses SOLWEIG and other "city-based climate services" [17].

5.3. Routing tool design

The routing tool has a modular structure, where separate components are all part of the workflow that make up the pedestrian route planner (see Figure 5.4). First, spatial input data were prepared for the selected study area, including the OpenStreetMap (OSM) pedestrian network and the input layers required by SOLWEIG. SOLWEIG was then run to produce the hourly UTCI maps. These outputs are linked to the pedestrian network by assigning environmental values from the raster maps to network edges. Finally, the routing component uses these weighted edges to calculate thermally conscious routes, which are made accessible through a web-based interface.

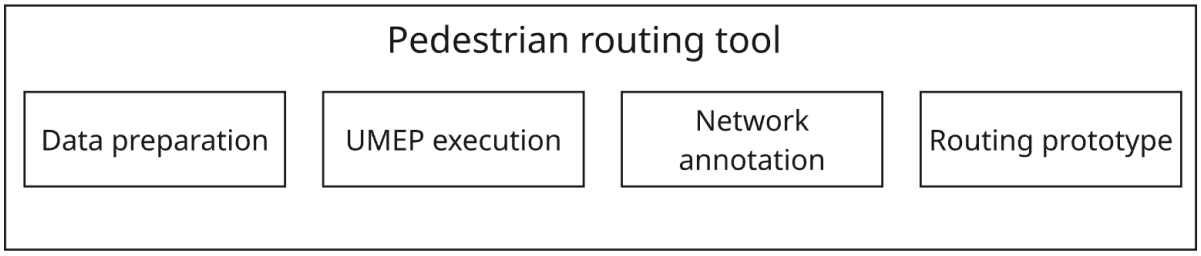

Figure 5.4: Overview of different tool modules.

The components of the tool were evaluated separately to make the insights generated by the thesis more broadly applicable. Existing research uses a wide variety of methods for thermally informed routing, which suggests that there are multiple valid ways of constructing such a tool. By evaluating each part of the workflow as a component-level design element, the findings can be interpreted independently of the complete implementation. This allows the results to remain relevant for other methodological contexts.

The description in the next sections represents the state of the research after the multiple design iterations. Where applicable the emergent design process is represented by tables showing initial approach, the bottleneck it introduced, and how the approach was revised. More information on the emergent process can be found in Appendix: Section A. For some elements the initial direction and approach gathered from literature proved sufficient and no revision was necessary to come to the final product.

Finally, the transferability of the tool was assessed by applying the full workflow to an additional study area in Rotterdam and evaluating its performance in terms of computation time and ease of use. Since a scalable tool should be easily transferable to a different spatial area.

5.3.1. Requirements

The central design requirement for the tool was scalability. Existing thermally conscious routing methods show that thermally informed routing is possible, but often remain limited to specific study areas. This makes it difficult to apply the same method to larger areas, different cities, or other environmental settings. In this thesis, scalability was therefore considered not only for the final route planner, but for every component in the workflow.

Firstly, performance was treated both as part of scalability and as a requirement in itself. In a computationally inefficient tool, existing bottlenecks are amplified as the tool scales. At the same time, performance was also important for usability. The routing component needed to generate routes within a reasonable time, especially when users interactively changed origins, destinations, timesteps, or preference settings. Additionally, faster components made iterative development, testing, and debugging more efficient, while also improving the scalability of the complete workflow. However, performance optimization was not treated as an objective in itself. Optimization decisions were weighed against their expected impact on the use case, in order to avoid over-engineering individual components where the practical benefit would be limited.

Secondly, ease of use was considered for all components. A scalable tool should not depend on extensive manual intervention, because manual steps make the workflow slower and more difficult to reproduce. In this context, manual intervention refers to tasks such as searching through separate code fragments, moving files between folders, adjusting file paths, or manually preparing inputs between processing steps. For this reason, the implementation aimed to minimize the manual intervention within the components.

Thirdly, reusability was addressed through a component-based architecture [15, 21]. The tool was divided into separate modules with low coupling[1] and high cohesion[2]. Each component had a clearly defined purpose and, where necessary, communicated with other components through explicit data formats. This made the components easier to test, replace, or extend without changing the entire workflow. Reusability also supported scalability, because future applications are unlikely to require the full methodology exactly as developed in this thesis. Depending on the available data, computational resources, and research aim, only specific components may be relevant. The modular design therefore allowed individual parts of the workflow to remain useful even when the complete route planner was not implemented.

Insight	Initial approach	Bottleneck	Revised
I9	Ease of use, performance, scalability	strong coupling between components introduces performance bottlenecks	modular system design with high cohesion that supports reusability

Table 2: Emergent process full tool

[1] Coupling is a term used in software design that describes how dependent software modules are on each other. High coupling means that changes in one module of the software can have big effects on other modules. Low coupling means that the modules are largely independent from each other.

[2] Cohesion is closely related to coupling and refers to how well all the elements in a module are supporting a single well-defined purpose. For reusability, low coupling and high cohesion are preferred.

5.3.2. Data preparation

5.3.2.1. UMEPIO

Monahan (2025) [23] developed SOLFD, which implements an automated pipeline for collecting the input data required to run SOLWEIG. SOLWEIG requires several spatial input layers, including Canopy Height Model (CHM), Digital Surface Model (DSM), Digital Terrain Model (DTM), and land-cover rasters. If these layers have to be collected and processed manually, the workflow becomes time-consuming, difficult to reproduce, and more vulnerable to human error.

However, the pipeline was distributed as separate sections of code in a GitHub repository [24], which limited its practical reusability. To address this, the existing SOLFD pipeline was published as a Python Package Index (PyPI) package: UMEPIO. Packaging the workflow makes it easier to install and reuse, while also standardising the way input data are collected and prepared. This reduces the amount of manual effort required within the data-preparation component and allows the same input-data pipeline to be reused in contexts beyond the route planner developed in this thesis.

5.3.2.2. Pedestrian network

Two main data sources were considered for constructing the pedestrian network. The first option was the OpenStreetMap, which is commonly used in routing research because it is open, widely available, and accessible through established tools such as OSMnx. The second option was Overture Maps, which provides cloud-native and normalised network data partly derived from OSM, and was therefore considered potentially preferable from a scalability perspective.

However, manual inspection of the study area showed that some less common tags were not retained in the Overture dataset. This was particularly relevant for pedestrian infrastructure, where sidewalk-related tags were missing. While sidewalk data in OSM are not complete or consistently available, the presence of these tags can still improve the spatial representation of the pedestrian environment. This is relevant because pedestrians do not use the centre lines of roads, but generally move along sidewalks or road edges. For this reason, retaining this information was considered more important than using a more normalised network source.

The final workflow therefore used plain OSM data, gathered through OSMnx [5][3], as the basis for the transportation network. The pedestrian network was prepared in a way that preserved information relevant to both routing and environmental annotation. This is important because the network is not only used to calculate paths, but also to link SOLWEIG outputs to the walking environment.

For this reason, a custom OSM filter was used instead of the standard OSMnx pedestrian filter [5], which removed some pedestrian information that was relevant for this use case. The network representation was also simplified by storing it as an undirected graph. Since pedestrian movement is generally possible in both directions, storing two directed edges for each node pair introduced redundant information and increased the number of edges that had to be annotated. Using a single bidirectional edge reduced storage requirements and processing effort while preserving the routing structure needed for the tool.

[3] OSMnx is a Python package that downloads OSM data into a NetworkX representation, allowing for easy access to the data.

Insight	Initial	Bottleneck	Revised
I1	Use the SOLFD pipeline directly from its original code distribution	The pipeline was difficult to install and run because of how the code was distributed	Distribute the SOLFD pipeline as a PyPI package
I3	Standard filtered OSMnx	Loss of tags	Personal pedestrian filtering
I3	Multiple ways to tag the same thing	routing requires flattened representation, but we don't want to lose any information	Add personal pedestrian tags
I4	Bi-directionality in standard representation enforced through double edges	Redundant information increases storage demand	Change network representation to undirected
I2	Centre lines	Pedestrians do not use the center lines of the road	Solution out of scope: provide analysis for future work

Table 3: Emergent process data preparation

5.3.3. SOLWEIG execution

For scalability running SOLWEIG introduced two main challenges:

- **Space:** SOLWEIG produces raster-based outputs for several environmental variables, including UTCI. For each timestep, one for every hour in the day, these outputs represent the entire study area as a raster layer. As a result, the total data volume increases rapidly when additional weather scenarios are included or when the spatial extent is expanded. This creates a scalability problem: storing the full raster based model output for every possible routing context would require large amounts of storage and would make the data slower to access during route calculation.
- **Time:** The computational cost of running UMEP simulations is substantial. Using SOLWEIG_GPU [14], processing a 2 km by 2 km area at a 0.5 m resolution required approximately 20–40 minutes, depending on the available hardware.

For these problems, two solutions were investigated. The first was a *generalist* approach, in which UMEP would be run on demand. This would reduce storage requirements, since the collection of input data and running of the model would be done only when a route is requested. However, this shifts the bottleneck to input-data collection. If the full tile surrounding the route of input data still needs to be collected and stored, there is little advantage over storing the model outputs directly. Alternatively, collecting smaller tiles around the route would require many requests to an external data server, such as PDOK. This would greatly increase latency and create a dependency on an external source during route calculation. For a scalable route planner, this is undesirable because it makes the system slower and less reliable.

The second solution was a *specialist* approach, in which UMEP outputs are precomputed for the study area and transformed before storage to become a manageable size. The computation time can then be moved to an external machine depending on hardware availability. This approach was chosen for this thesis. SOLWEIG_GPU was run on the DelftBlue supercomputer

which has multiple GPUs shared across the university. The performance (computation time) of SOLWEIG_GPU was entirely dependent on the RAM available. With tests on the same GPU having either 10GB or 80GB of RAM produces computation times for the test area of 37 minutes or 18 minutes respectively. These large differences indicated that a full performance benchmark would not be additive to this research, since any further work would most likely have access to different hardware anyway. Tile size was not tested but determined using the recommendations provided by the developers of SOLWEIG_GPU [14].

The output of SOLWEIG_GPU consists of multiple partially overlapping raster tiles, which must be merged to reconstruct a continuous raster for the full study area. The overlap between adjacent tiles is specified as an input parameter during the SOLWEIG_GPU simulation. During the merging process, this overlap is divided between neighbouring tiles: each tile is cropped along its internal edges by half of the overlap distance before the remaining raster sections are joined. This prevents duplicate representation of overlapping cells while preserving the full spatial extent of the modelled area.

To assess the computational implications of spatial resolution, the raster data were processed at multiple resolutions and the corresponding runtimes were recorded and compared.

Insight	Initial	Bottleneck	Revised
I5	Run umep on demand on small area	Gathering input data too slow	methodological approach abandoned
I6	Use external compute infrastructure	Performance depended strongly on available hardware	Do tests on the available hardware before defining the workflow

Table 4: Emergent process UMEP execution

5.3.4. Network annotation

This component links the SOLWEIG outputs to the pedestrian network by transforming raster-based UTCI data into edge-level network attributes. The SOLWEIG output represents thermal conditions as continuous raster surfaces for each timestep, whereas the routing algorithm operates on a graph in which costs are assigned to individual edges. Therefore, the raster values cannot be used directly during route calculation. They first need to be aggregated into a single representative value, for each network edge. This transformation converts the spatially detailed UTCI output into a graph-based representation that is compatible with the routing algorithm.

To achieve this, the SOLWEIG raster outputs were sampled along the geometries of the pedestrian network and transformed into edge-level attributes. For each edge, the raster cells intersected by its geometry were identified, after which the UTCI values from each of the 24 raster bands were sampled. The values of all intersecting cells within a single band were then aggregated by taking the median, resulting in one representative `utci_median` value per timestep. These median values were mapped to their corresponding UTCI thermal-stress categories and stored on the edge as the attribute `utci_category` Table 5. This compact edge-level representation allows the spatially detailed raster output to be incorporated directly into the edge weights used by the routing algorithm.

To make the sampling process efficient, the SOLWEIG output was converted to a file format better suited to repeated spatial access. The merged SOLWEIG output was stored as a Zarr file, a cloud-optimized format that supports efficient access to multidimensional raster data

(see Section 6.4.2). Compared with the previous GeoTIFF-based workflow, this reduced the network annotation time from approximately five minutes to around 20 seconds on first access, and to about 10 seconds when the data was already cached.

The algorithm does for every edge in the graph $G(V, E)$:

1. Get the geometry of the edge.
2. Identify the raster cell indices of all cells intersected by the edge geometry.
3. Extract the UTCI values from all 24 raster bands for the intersecting cells.
4. For each raster band, aggregate the sampled cell values into a single representative edge value.
5. Store the resulting 24 values as an edge attribute.

UTCI (deg C)	Thermal stress
> 46	Extreme heat stress
38 to 46	Very strong heat stress
32 to 38	Strong heat stress
26 to 32	Moderate heat stress
9 to 26	No thermal stress
0 to 9	Slight cold stress
-13 to 0	Moderate cold stress
-27 to -13	Strong cold stress
< -40	Extreme cold stress

Table 5: UTCI categories as defined for SOLWEIG [6] .

Where the categories in Table 5 represent increasingly high penalties to the edge weight. This is in line with findings from Basu et al. (2024), who found that higher UTCI categories have "non-uniform and possibly exponential" effect on perceived walking distance [4]. Using categories of an outdoor thermal comfort index to assign edge weights is consistent with the methodological approach of Ma et al. (2025), who applied Physiological Equivalent Temperature (PET)-based categories for this purpose [18]. In this thesis, UTCI was used instead of PET due to the absence of detailed wind information. Nevertheless, the underlying principle remains comparable, as both approaches translate thermal comfort categories into routing penalties.

The annotated network is saved as two GeoParquet files, one containing the edges and one containing the nodes, using a shared custom data schema. This schema acts as the translation layer between the network representation used by the annotator and the router, as shown in Figure 5.5. In addition to improving consistency between both components, this representation also reduced storage requirements. The original OSMnx GraphGML file required approximately 26 MB, whereas the equivalent GeoParquet representation required only about 1 MB.

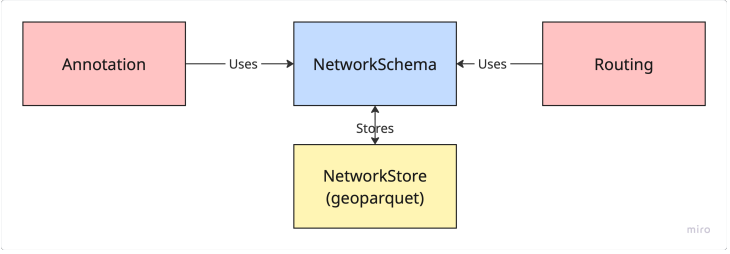

Figure 5.5: Shared network representation

Insight	Initial	Bottleneck	Revised
I7	Sample edge geometry from merged tif	Multiple lookups for every band negatively impact performance	Merge tif tiles in a .Zarr format
I8	Use OSMnx networkX representation	Introduces performance issues during routing	Shared data schema and storage in geoparquet, storing nodes, edges and metadata

Table 6: Emergent process network annotation

5.3.5. Routing Prototype

This component uses the annotated pedestrian network both for visualization in the front-end and for applying the routing algorithm in the back-end. The front-end and back-end communicate through an Application Programming Interface (API), which supports the broader requirement of reusability. Separating these components prevents the routing logic from becoming dependent on the interface. This is especially relevant at larger spatial scales, where executing routing operations directly in the front-end could introduce performance constraints.

The network is loaded from the shared `NetworkSchema` and converted into a node-edge representation using NumPy, while maintaining an adjacency list for efficient graph traversal. This representation was selected because it is conceptually transparent and computationally efficient for pedestrian routing tasks.

In line with previous work the algorithm is an implementation of weighted Dijkstra [2, 3, 18, 35, 39]. With the algorithm having the following two characteristics:

1. The algorithm should be dynamic, meaning that edge weights are not fixed in advance but depend on the pedestrian's state as they reach each node. This state is shaped by the intensity and duration of prior heat exposure during walking and can therefore influence the weighting of subsequent edges, as demonstrated by Wen et al. (2025), Rußig and Bruns (2017), and Zhao et al. (2024) [35, 39, 42].
2. The routing process should not rely solely on researcher-defined weights, as this would impose fixed assumptions about the relative importance of environmental factors. Instead, it should incorporate user-defined environmental preferences, similar to the approach proposed by Novack et al. (2018) [27].

The first requirement reflects that experienced thermal discomfort is cumulative and dynamic. Heat stress is not determined only by conditions on individual street segments, but also by the intensity and duration of exposure accumulated along the route. The routing algorithm therefore carries a route state through the search, representing the pedestrian's accumulated heat exposure at each node. As a result, the same physical node can be reached under different thermal conditions.

The algorithm therefore stores costs for combinations of node and route state, rather than only one least cost per node as in a standard shortest-path implementation. This allows accumulated heat exposure to influence subsequent edge costs, which is implemented through the dynamic sensitivity function. When pedestrians traverse multiple thermally stressful edges consecutively, this function increases the penalty assigned to further heat exposure. This is consistent with Zhao et al. (2024), who show that thermal sensation is influenced not only by

current conditions, but also by prior heat exposure [42]. Consequently, the algorithm increases the relative importance of cooler or shaded alternatives after sustained heat exposure.

The second requirement is motivated by the usability of the tool. Instead of only imposing fixed researcher-defined trade-offs between route length and thermal comfort, users are able to indicate their own preferences. This allows the routing behaviour to adapt to different pedestrian needs.

5.3.5.1. Edge cost calculation

Each edge in the pedestrian network receives a generalized cost:

$$C_e = L_e \cdot (1 + P_e)$$

where C_e is the edge cost, L_e is the edge length in meters and P_e is the normalized environmental penalty multiplier. This formulation ensures that route length always remains part of the optimization problem. For example, an extremely hot but short edge may still be preferable to a longer detour at medium exposure to heat.

The environmental penalty is defined as:

$$P_e = \sum_{i=1}^{n} w_i * p_i * \gamma_i(s)$$

where w_i represents the user-defined importance of environmental variable i, p_i is the normalized environmental penalty associated with edge e and variable i, and $\gamma_{i(s)}$ is a dynamic sensitivity function dependent on the route-state s.

The environmental penalty scales with the severity of the heat stress [4]. These multipliers effectively increase the distance of hot edges, which makes the algorithm more likely to choose a slightly longer route that is more comfortable.

Heat stress	**Multiplier**
No thermal stress	0.00
Moderate heat	0.10
Strong heat	0.35
Very strong heat	1.00
Extreme heat / near-avoid	3.00

Table 7: Heat stress categories and their associated penalty multipliers used in the routing algorithm.

The application was developed in a fully Python-based environment as a tool for communicating and testing the routing logic. Since numerous front-end frameworks and implementation strategies could have been used, detailed exploration of front-end development was considered outside the scope of the research. Due to its limited relevance to the research question. The front-end was therefore deployed as a prototype, while remaining separated from the routing logic. This ensures that the routing component and the broader processing workflow remain reusable for other applications.

The interface uses a MapLibreGL map, on which routes can be generated by clicking two points on a map and setting the preferences.

Insight	Initial	Bottleneck	Revised
I8	NetworkX representation	Performance issues at scale	Custom network representation
I9	Application also does routing	Coupling of elements can negatively effect reusability and client-side calculation of routes introduces performance constraints	Separate routing from the application and communicate between the two via an API

Table 8: Emergent process routing network

5.4. Validation and evaluation of the route planner

In this thesis there is a distinction between validation and evaluation. Validation refers to testing the route planner's output in a structured way. This is done by examining how much UTCI exposure decreases compared with the shortest route, how much additional distance this requires, and how spatially different the resulting thermal-comfort routes are from the original shortest-distance routes. While evaluation encompasses the reflexive method used to generate the generalizable insights from the design iterations. The evaluation method was described in Section 5.1 and an overview with the insights can be found in Appendix: Section A.

The validation was performed as follows:

1. **Manual validation:** Two routes with a large difference between the shortest and most comfortable path were manually evaluated. This evaluation is done using the original SOLWEIG output maps to validate that the conversion from raster-based outputs to edge-level attributes is not causing any meaningful data loss or changes. This method is less structured, but aligns with how validation was happening naturally during the development process.
2. **Structured validation:** This method was used for final validation, but also for tuning the variable weights and attribute choices in the algorithm. For this purpose, a stratified origin-destination (OD) test set was created. The test set consists of randomly sampled OD pairs from the pedestrian network, grouped into distance-based categories. The distance between an origin and destination is calculated as a straight-line distance: geodesic distance is used when the coordinates appear to be longitude-latitude coordinates, while Euclidean distance is used otherwise. The maximum possible OD distance is estimated from the bounding box of the node dataset.

For every category 100 OD pairs were generated where the directions were also randomized. Plot of OD pairs in Figure 5.6.

- `short`: 100 to 400 meters.
- `short_mid`: 400 to 750 meters.
- `medium`: 750 to 1100 meters.
- `medium_long`: 1100 to 1500 meters
- `max_distance`: above 1500 and between the estimated max available distance based on the bounding box.

Figure 5.6: The five categories of OD pairs. With the green cross denoting the origin and the orange circle denoting the destination.

This validation serves more as a sensibility test than a true validation of thermal comfort along the routes. It aims to show that the thermally comfortable routes are notably different from the shortest path; in distance and average temperature. But also that the algorithm behaves in a predictable way for different types of routes.

6. Implementation

This chapter describes the technical implementation of the tool. Discussing the data structures and functions used in the code as well as the structure of the entire tool. Figure 6.1 shows the connection between the separate components in a graphical manner.

First, the chapter will present the implementation rationale for the entire tool. Then it wil go into technical detail for all the components. The structure of the chapter is based on the structure of the code, which differs in some aspects from the conceptual separation used in Section 5. For example, the configuration of the pedestrian network is not included in the data preparation section since it is happening in a different section of the code. The goal of this chapter is for a developer to be capable of re-building the tool.

6.1. Full tool

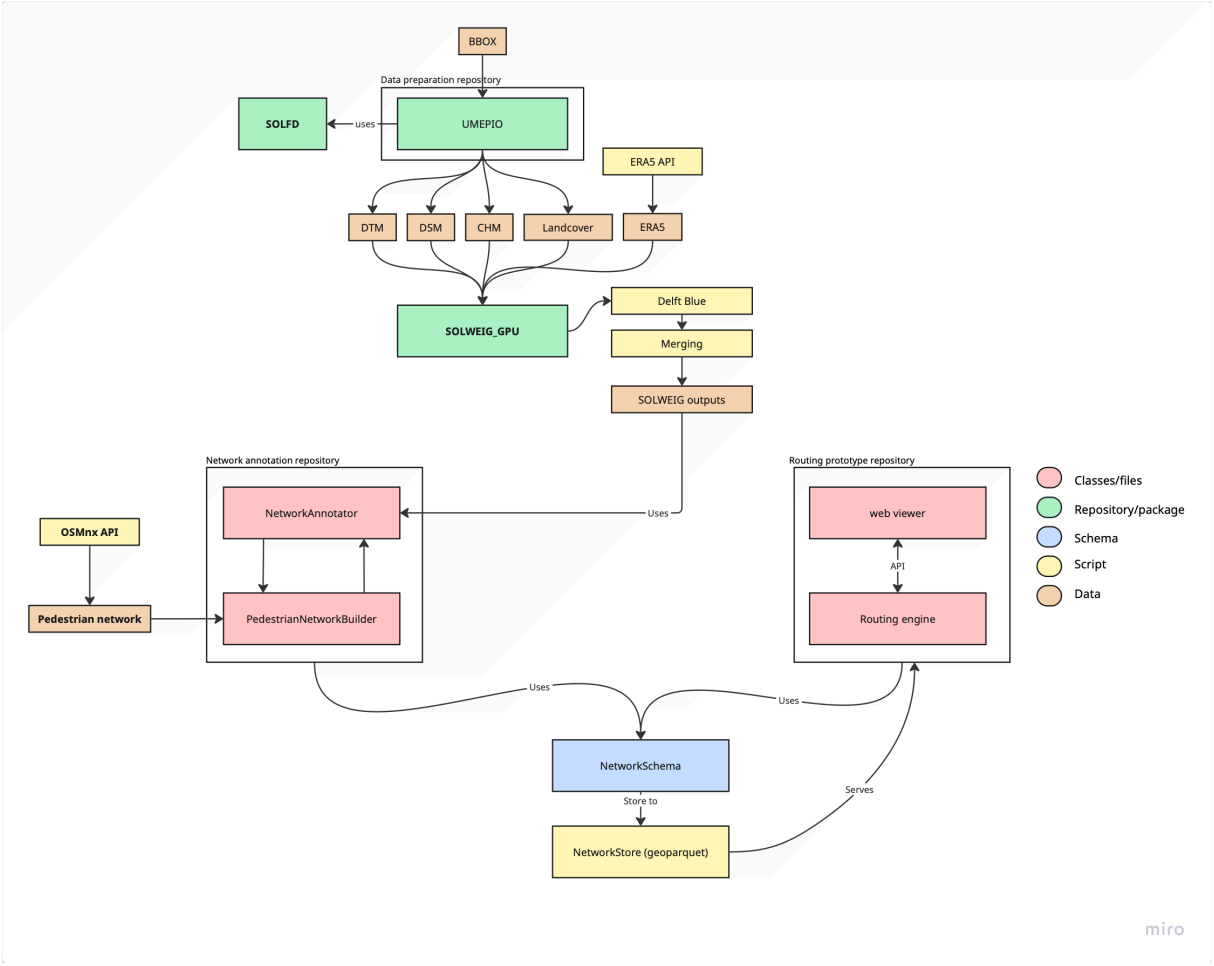

Figure 6.1: Graphical representation of the connection between the components.

The pedestrian routing tool uses a component based architecture (CBA), which emphasizes modularity and is associated with a better reusability [15]. This is technically enforced through the active separation of code according to Figure 6.2.

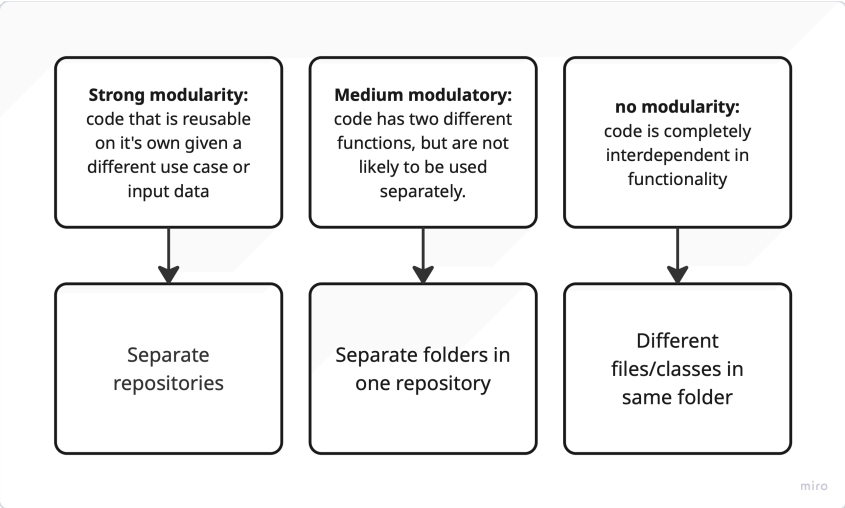

Figure 6.2: Code structure to ensure modularity.

6.2. Data preparation

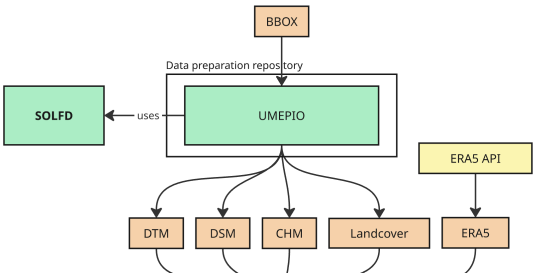

Figure 6.3: Structure of the data preparation code.

6.2.1. UMEPIO

UMEPIO, like SOLFD [24], is organised around four internal classes: buildings, DEMs, CHM, and landcover. With the classes reflecting the main input categories required by SOLWEIG. The building class retrieves and stores building geometries for the study area. The Digital Elevation Model (DEM) class generates aligned terrain and surface rasters, using the building geometries where needed to represent urban form. The CHM class derives vegetation height information from LiDAR data and converts this into a raster layer. The land-cover class classifies and rasterises surface types so that they match the spatial grid of the other inputs. More detailed descriptions on the processing steps can be found in and Monahan (2025) [23].

Access to the functionality is organised through a user-facing API module, which uses the internal classes and exposes their use through simple callable functions (Figure 6.4). Users can either run separate steps, such as only generating buildings or land cover, or call one high-level function that runs the complete input-generation workflow. This keeps the original processing logic modular while making it easier to (re)use.

Lastly, due to a bug in SOLWEIG_GPU where it is unable to consider negative values all the negative values of the generated DEMs are set to 0. Note that this is to account for a bug and not desired behaviour which is why it is not implemented in UMEPIO directly.

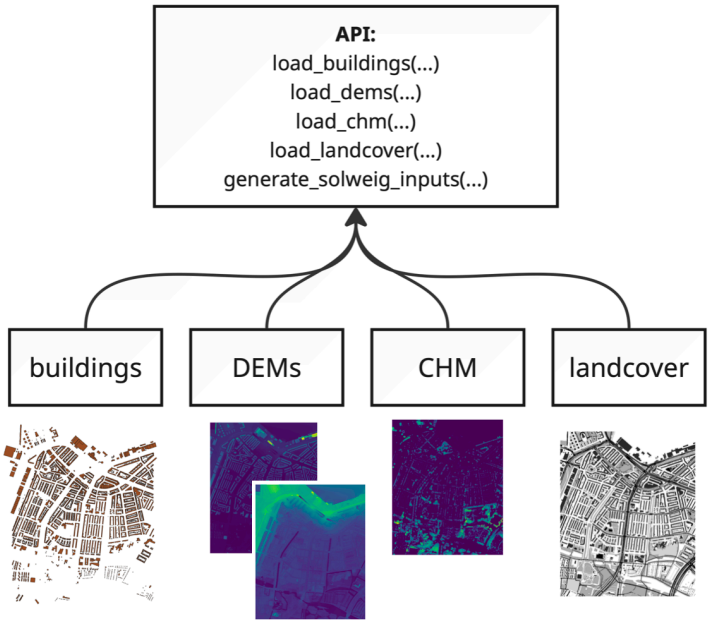

Figure 6.4: Structure of UMEPIO package.

6.3. Running SOLWEIG

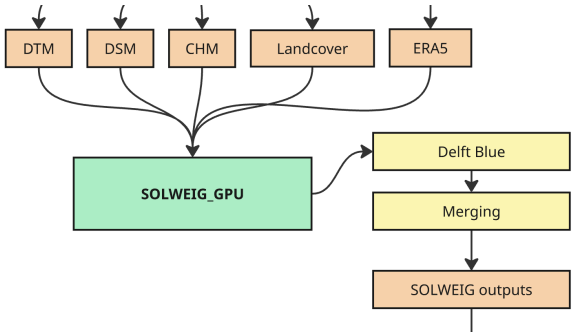

Figure 6.5: Structure of collection of scripts used for running SOLWEIG

SOLWEIG_GPU was run on DelftBlue, the TU Delft high-performance computing cluster shared with the entire university. DelftBlue provides CPU, high-memory, and GPU nodes, including GPU partitions with NVIDIA V100 and A100 GPUs, making it suitable for computationally intensive raster-based modelling tasks [7]. The model was executed through the SOLWEIG_GPU package [14, 37], using a single function call to run the simulation after the required input rasters and meteorological files had been prepared.

Running on DelftBlue required some adaptation compared with running the model locally. Since only the login nodes have a direct internet connection, the generation and downloading of input data was performed locally before transferring the prepared files to DelftBlue. On DelftBlue, SOLWEIG_GPU was loaded in a Python virtual environment and executed through an `sbatch` script, which submits the job to the `Slurm` queue until the requested resources become available. Although GPU nodes with higher RAM achieve faster runtimes, the small GPU partition with 10GB of RAM was often more practical because there was no queue. As a result, the total time between submitting the job and obtaining the output could be shorter, even when the runtime itself was slower. SOLWEIG_GPU automatically tiles the input data before

processing based on a user-defined tile size; since the documentation states that the optimal tile size depends on available GPU memory [37], tiles of 1000 × 1000 cells were used.

6.4. Network annotation

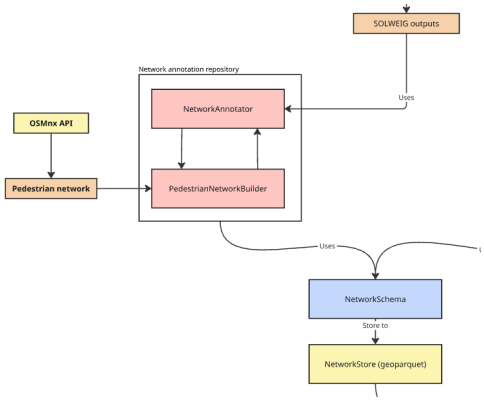

Figure 6.6: Structure of network annotation code

6.4.1. Network builder

The network is retrieved from OSM [30] using OSMnx [5]. A custom filtering step is then applied in Python to select the relevant edges. The filtering keeps only edges that are usable for pedestrians. In practice, an edge is excluded when it is clearly unsuitable according to the OSM documentation, for example `highway=motorway` or `access=private`. Edges are included when they are suitable for pedestrians, such as `highway=footway`. Roads that are not explicitly pedestrian-only, such as `highway=residential`, are also included. Since explicit pedestrian information is relatively rare, edges with such information are given an additional custom tag, `ped_infra_type=dedicated`, to reflect this. When sidewalk information is available for certain roads, it is also added through the tags `sidewalk_present` and `sidewalk_side`.

6.4.2. Network annotation

For every edge in the network, the UTCI raster is sampled along the edge geometry. The SOLWEIG output is stored as a multiband Zarr raster, where the spatial data are chunked and each chunk contains all 24 hourly bands (Figure 6.7). In this implementation, the chunk structure is `256x256x24`, which means that the original multiband GeoTIFF is reorganised into chunks of 256 by 256 cells, while all 24 hourly UTCI bands are stored on top of each other within each chunk. The conversion to Zarr therefore does not change the raster values, resolution, or spatial extent, but changes how the data are stored and accessed.

Because network edges are spatially clustered, annotating nearby edges is likely to access the same or adjacent Zarr chunks, limiting the amount of data that needs to be loaded into memory. Furthermore, since all 24 hourly bands are stored within each spatial chunk, the complete daily UTCI profile for a raster cell can be retrieved in a single read operation. This enables each edge to be assigned a 24-value thermal attribute without repeatedly accessing separate raster files or individual bands.

All these values are collected and aggregated into two attributes: `utci_median`, which stores the median UTCI value per raster band, and `utci_category`, which stores the corresponding thermal-stress category. Lastly, the names of the added environmental attributes are stored in the network metadata.

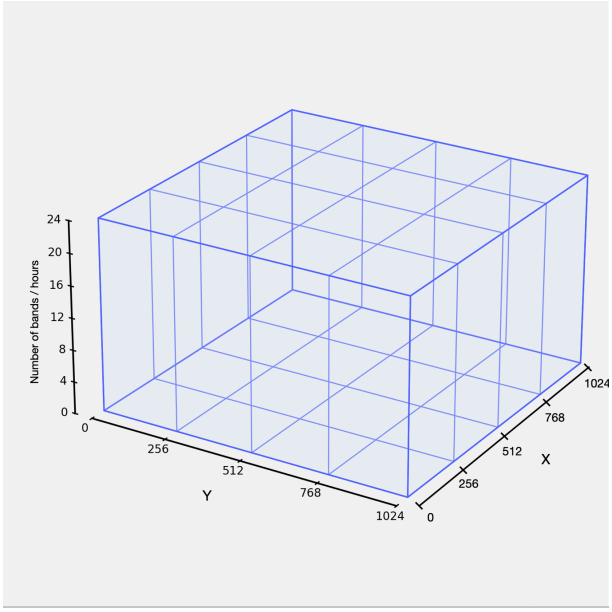

Figure 6.7: Chunked access to the zarr cube adjusted from Trujillo et al. (2026)[38]

6.4.3. Network storage

During processing the network is kept in the NetworkX representation used by OSMnx, namely `nx.MultiGraph` combined with custom metadata. Using the combination of both the network can be visualized and loaded/saved from/to persistent storage[4]. The data schema simply defines which columns need to be kept so the network can be converted reliably between different network representations. From the runtime representation to persistent storage the network is written to two GeoParquet files and one metadata file:

- `nodes.parquet`: contains the original OSM node id and the geometry `point(x, y)` of its location.
- `edges.parquet`: contains the origin and destination node, the OSMnx key, all annotated attributes, and any remaining attributes preserved after filtering.
- `metadata.json`: contains the metadata assigned throughout the annotation process, such as the CRS and the names of the annotated attributes.

[4] In the case of this thesis, persistent storage just means that the network is saved to a file (geoparquet). However, depending on the implementation, other forms of persistent storage, such as a database, could also be used.

6.5. Routing prototype

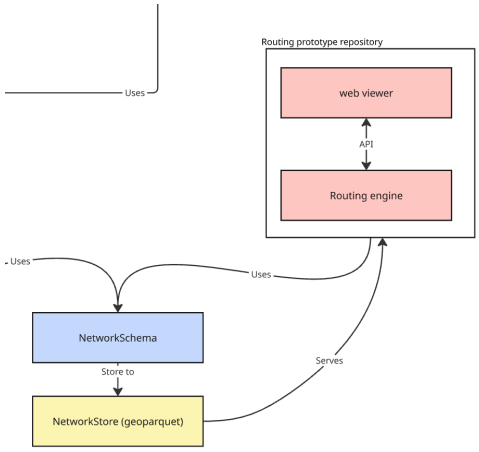

Figure 6.8: Structure of the routing prototype

The routing prototype connects the prepared pedestrian network, the routing algorithm, an API layer, and an interactive web interface. The prepared network is loaded from the persisted schema folder containing the node table, edge table, and metadata. At startup, this network is converted into the routing representation used by the algorithm and is also transformed into a GeoJSON payload that can be displayed in the browser. This means that the same underlying network is used both for visualisation and for route calculation, avoiding a separation between what the user sees and what the algorithm uses.

6.5.1. Adjacency list

For efficient access during routing, the network topology is represented through an adjacency list. This structure stores for each node, the neighbouring nodes that can be reached from it (see Figure 6.9). Each neighbour entry also stores the corresponding edge. This is important because the neighbour relation itself only describes connectivity, while the edge position provides access to the attributes needed for environmental cost calculation. The adjacency list is created when the network is initialized for routing. So during routing all connections from a single node can be retrieved with a single query. Because the pedestrian network is undirected, this initialization step also inserts each edge in both directions, allowing all movement between connected nodes.

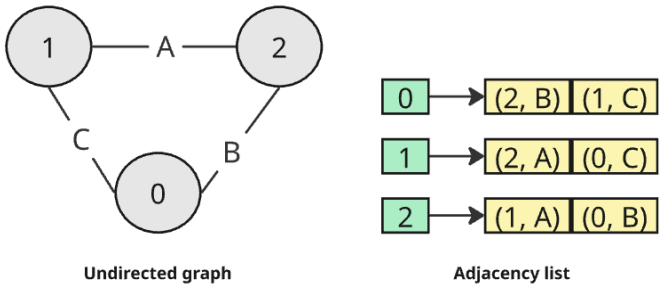

Figure 6.9: Adjacency list representation of an undirected graph. Adjusted from GeeksforGeeks [11].

6.5.2. Modified Dijkstra

Standard Dijkstra is explained in Section 3. This implementation follows the same principle, but tracks the least-cost path for each combination of node and route state rather than for each node alone. This is necessary because future edge costs depend on both the current location and the accumulated heat exposure at that point.

The algorithm uses a priority queue to always expand the currently cheapest partial route. Each queue entry contains the current accumulated cost, the current node, and the current route state. When an entry is selected from the queue, the algorithm checks all neighbouring nodes. For each possible next edge, it calculates the weighted edge cost and the updated route state. The new total cost is then compared with the best known cost for the resulting node-state combination.

If the new combination of (node, state) has not been reached before, or if it has now been reached with a lower cost, it is added to the queue. The algorithm also stores the predecessor of each accepted node-state combination. This is required because the final route must be reconstructed after the target has been found. The search stops when the target node is reached. At that point, the algorithm has found the best route under the implemented cost structure, including both distance and dynamic thermal penalties.

The final route is then transformed back to its original OSMnx IDs. This allows the retracing of the route in the front-end, since the internal network representation for routing is not used by the front-end.

6.5.3. Application

The application is structured as a local client-server prototype, with the backend API and Shiny interface served separately. This keeps the routing logic outside the interface code: the Shiny application handles map interaction and visualisation, while the API manages the network, route requests, algorithm execution, and response formatting.

Routes are generated through direct map interaction using MapLibreGL. The first click sets the origin and the second click sets the destination. These coordinates, together with the selected hour, are sent to the API, which snaps them to the nearest network nodes, calculates the route, reconstructs the traversed edge geometries, and returns the result as GeoJSON for visualisation in the interface. This is shown in Figure 6.10.

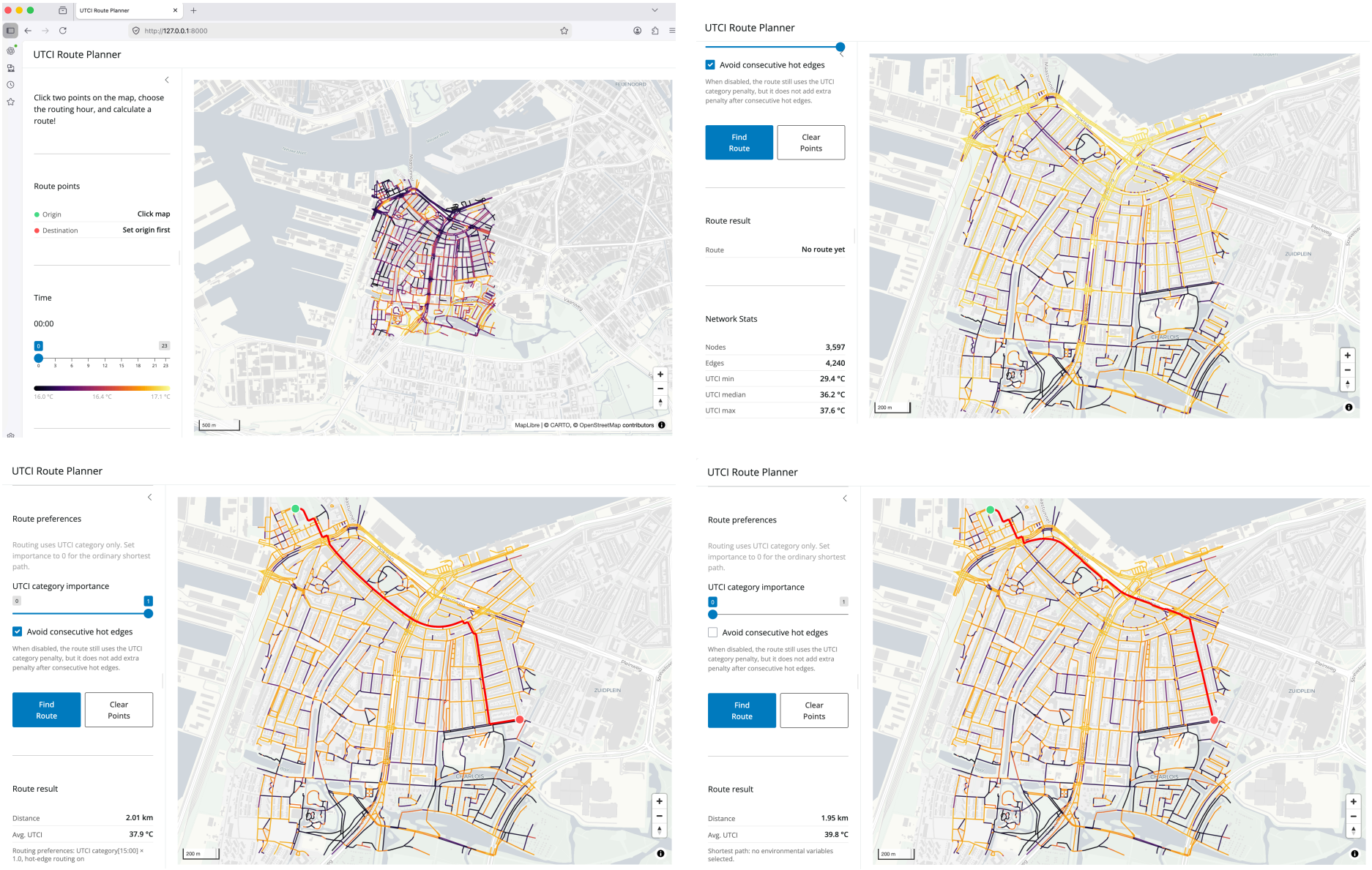

Figure 6.10: UI at start-up (top-left), UI when time is set to 12:00 (top-right), UI when comfortable route between OD is chosen (bottom-left), UI when shortest route between OD is chosen (bottom-right).

7. Results and analysis

This chapter presents the main results of the developed workflow and analyses what they reveal about thermally conscious pedestrian routing. It first reflects on insights that emerged during the development process, before evaluating the implemented tool through its performance and routing outcomes. The detailed emergent design process is summarized in the Appendix: Section A.

7.1. Generalizable insights from the emergent process

This section synthesizes the main lessons from the emergent development process. Rather than repeating each design decision from the methodology chapter, it focuses on what the implementation revealed about the conditions under which a thermally conscious pedestrian routing tool can be scalable, reusable, and computationally feasible.

7.1.1. Scalability through ease of use

During input-data preparation it became visible that scalability (in part) depends on the reusability and ease of use of the supporting workflow. Since reducing manual steps and standardizing execution were necessary before the SOLFD [23] workflow could be reused reliably. Publishing the data-preparation pipeline using PyPI UMEPIO therefore contributed to scalable transformation of SOLWEIG input data by making the preceding environmental-data workflow easier to reproduce, automate, and apply in other contexts.

7.1.2. Pedestrian network assumptions

The preparation of the pedestrian network showed that network extraction and representations are not neutral steps in the process. The environmental data is ultimately linked to the geometries of the pedestrian network, but these geometries are themselves abstractions. In this thesis, the network is based on OpenStreetMap geometries, which often represent movement through centre lines. Which is appropriate for car travel, but these centre lines do not necessarily correspond to actual walking paths, especially in areas where pedestrians would move on sidewalks or squares.

This matters because the thermal conditions assigned to a network edge depend on where that edge is geometrically located. If the geometry does not represent the actual pedestrian position, the sampled environmental conditions may also differ from the conditions pedestrians experience in reality. This does not make the network unusable, but it does mean that pedestrian network preparation should be treated as a methodological decision rather than as a purely technical input step. Figure 7.1 shows which parts of the network have tags that are related specifically to pedestrian use.

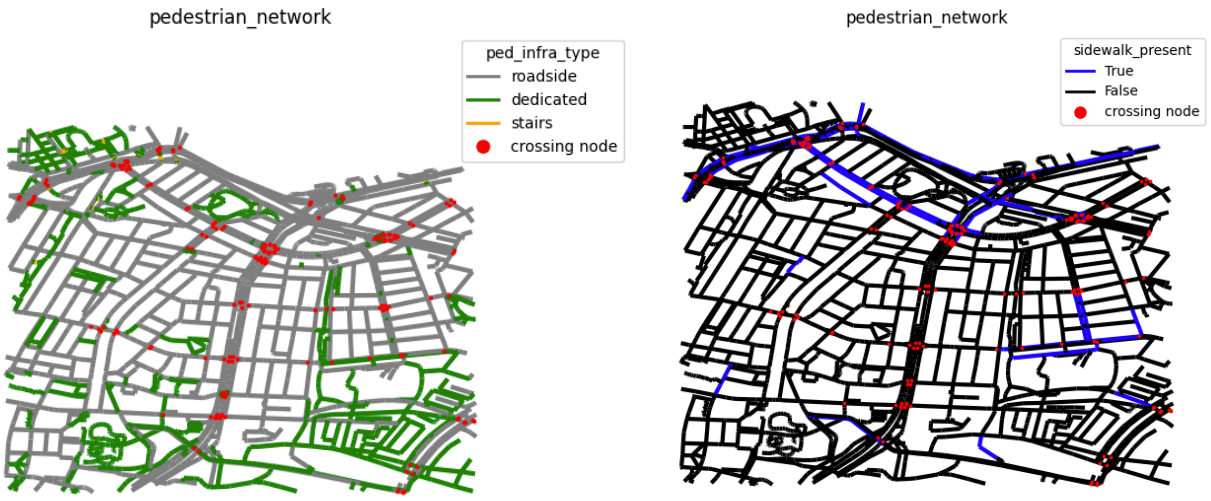

Figure 7.1: (Left) Green edges represent edges that have tags denoting them as dedicated pedestrian infrastructure. (Right) the blue edges represent segments that have explicit sidewalk tags.

The development process also showed that automatic filtering of OpenStreetMap data can introduce unnoticed data loss. Existing download tools (like OSMnx [5]) simplify network collection, but they may remove sidewalk and footway information. Losing information is not necessarily problematic when it is intentional and documented. However, since the filtering is hidden inside an abstraction layer, the final user may not know which information has been removed or why. This means that the choice of software used for data collection can influence the final routing result in ways that are not immediately visible.

Another practical insight concerns the representation of directionality. The standard NetworkX representation stores bidirectional movement using duplicate directed edges, one edge $u \rightarrow v$ and one $v \rightarrow u$. Pedestrian movement can be treated as undirected, so this representation creates redundant information. It doubles storage demand and processing time during annotation, because all connections are considered double. This shows that standard network representations should be questioned when their assumptions do not match the use case. A representation that is useful for general network analysis is not necessarily the most efficient data structure for scalable pedestrian routing.

7.1.3. SOLWEIG computation

Running SOLWEIG on demand is not feasible for interactive routing using the current workflow. Initially, running SOLWEIG only for the area directly required by a route seemed attractive, because it would avoid generating environmental data for unused areas. In practice, however, this approach was bottlenecked by input data collection process. Downloading and merging the required tiled input data for all data sources introduced too much latency for route-time computation. As a result, SOLWEIG outputs need to be precomputed and transformed before routing, rather than generated during route calculation.

This workflow is well suited to external compute infrastructure. However, the performance of SOLWEIG_GPU depends strongly on the specific computing environment, including available GPU memory, tile size, and shared workload conditions. In practice, each implementation should test SOLWEIG_GPU on the available hardware before drawing conclusions about the computation time needed to run SOLWEIG_GPU for a large area.

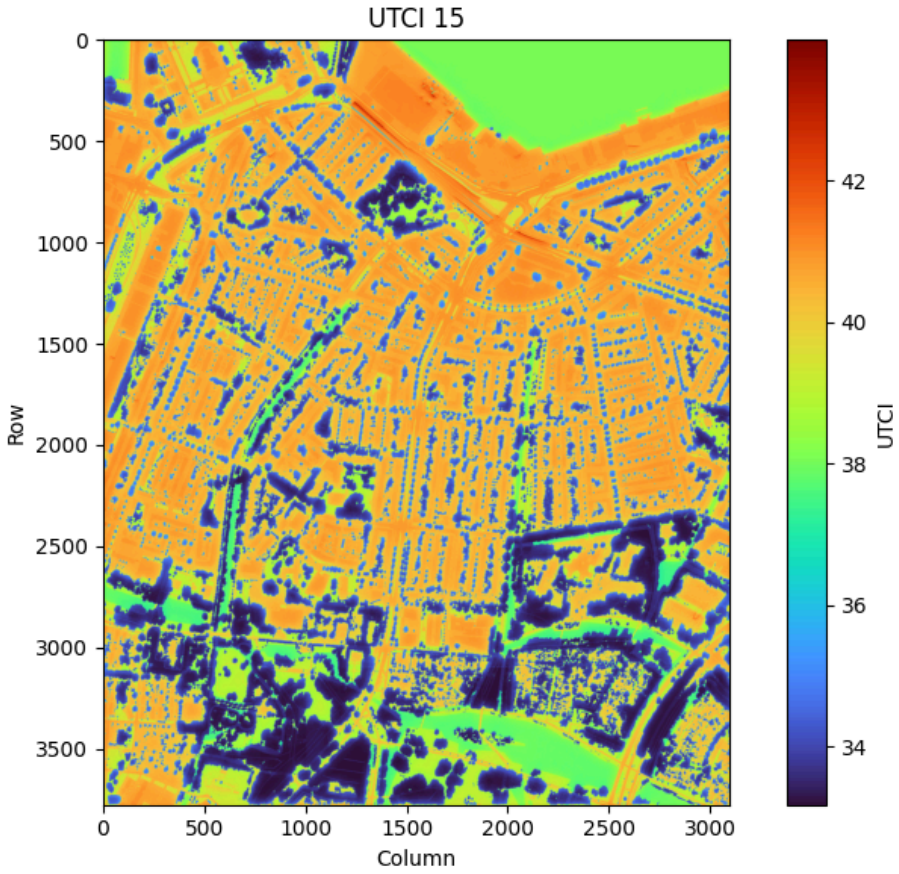

Figure 7.2: The UTCI of the study area at 15:00.

7.1.4. SOLWEIG integration in the pedestrian network

Integrating environmental information into the pedestrian network showed that file format choices can have a large effect on performance. This was especially visible when comparing multidimensional GeoTIFF storage with a Zarr-based structure. While a GeoTIFF can store spatial data, it was less suited to the repeated spatial and temporal lookup required during network annotation. Zarr better matched this access pattern and therefore improved network sampling performance on the study area from 5 minutes to 10 seconds.

This illustrates a broader point: file formats should be chosen based on how the data will be used, not only on whether they can store the required information. In this workflow, the environmental data are not visualized as maps; they are repeatedly queried to annotate many network edges. This is especially relevant for geodata, where the location of each value is part of its meaning and data access is often spatially organized. A format that supports repeated spatial and temporal lookup can therefore provide substantial performance gains.

The same principle applies to the storage of the pedestrian network. Using an explicit column-based format such as Parquet improved clarity because it required the network to be stored as a structured data table rather than as a fixed graph object. This separation between persistent storage and internal routing representation made the workflow more modular. Each component could transform the shared data into the structure best suited to its own computation. Thus, the chosen file formats can either support or constrain flexibility and performance when integrating SOLWEIG outputs in the pedestrian network.

7.1.5. Routing implementation

A routing-specific internal representation based on simplified NumPy arrays was used. In this representation, a single thermally conscious route took approximately 0.0364 seconds on average to compute. This suggests that, once the environmental data have been prepared and incorporated into the network, the routing calculation itself is not the main computational bottleneck.

The separation between the routing backend and the application was important because they serve different functions. The backend handles data access, route calculation, and thermal comfort costs, while the application focuses on interaction and visualization. Keeping these responsibilities separate improved the cohesion of the components. The communication between these two using an API supports reusability since the routing backend can be accessed by different interfaces without changing the underlying routing logic.

The development process also showed that the research value of the front-end was more limited than initially expected. For this thesis, the front-end mainly needed to demonstrate that the routing algorithm could be used interactively and that the results could be visualized.

7.2. Evaluation of the tool

7.2.1. Performance of the implemented workflows

The runtime tests in Table 9 should not be interpreted as a structured benchmark, but as an indicative comparison of the relative computational cost of the different workflow components on the available hardware. SOLWEIG was executed on DelftBlue using an A100 GPU with 10 GB of available GPU memory, while the remaining steps were executed locally on a MacBook Air with an M2 chip. The purpose of these tests was therefore not to establish general performance values, but to show where the main computational bottlenecks occur within this implementation and how changes in spatial resolution affect each step of the workflow.

Data set	UMEPIO	SOLWEIG execution	Network annotation
Study area (resolution 0.5 meter)	18.12 min	39.17 min	10 seconds
Study area (resolution 1 meter)	13.93 min	8.50 min	6 seconds
Study area (resolution 1.5 meter)	14.38 min	4.97 min	5 seconds
Different area	9.56 min	26.59 min	8 seconds

Table 9: Runtimes for different resolutions

The results show that the preparation and simulation of the SOLWEIG data are the most computationally demanding parts of the workflow. In particular, SOLWEIG execution time decreases substantially when the spatial resolution is reduced. This is expected, because a coarser resolution reduces the number of raster cells that need to be processed during the simulation.

The test on a different area was included to assess whether the workflow could be applied outside the original study area without requiring changes. No implementation issues were encountered: the new bounding box could be inserted into the same data preparation, SOLWEIG execution, and network annotation steps. The calculation times were slightly lower than for the original study area, which is caused by the smaller spatial extent of the second area.

The effect of spatial resolution on the quality of the thermal outputs was not validated in this thesis. From a computational perspective, lowering the resolution clearly improves performance, especially for SOLWEIG execution. Yet a coarser resolution also reduces the spatial detail of the simulated microclimate conditions and therefore affect the accuracy of the thermal attributes assigned to the pedestrian network. Without validation against observed thermal comfort data or human thermal perception, it is not possible to determine whether the performance gains from lower-resolution simulations justify the potential loss of accuracy.

7.2.2. Route validation

The route validation uses a generated set of stratified Origin-Destination (OD) pairs (Section 5: Figure 5.6). The two forms of validation are described in this section: manual and structured. The goal of the manual analysis is to circle back to the initial input data, since this was flattened into the pedestrian network to a single value per edge, even though the edges can have a high variability along themselves. The goal of the structured validation is to check the average performance of the algorithm across different route lengths.

7.2.2.1. Manual

For the manual analysis two routes were chosen from the stratified OD pairs (see Figure 7.3). These routes were in the top 25% of routes in terms of UTCI reduction, and from those the two routes with minimal difference in distance. This represents the optimal use-case for the routing tool, since it shows the highest gains (UTCI reduction) for the minimal effort (additional distance). OD pair 229 has some shared edges, but diverges twice to use more thermally comfortable sub-paths. OD pair 408 follows an entirely different path for the shortest and thermally comfortable routes, with no shared edges.

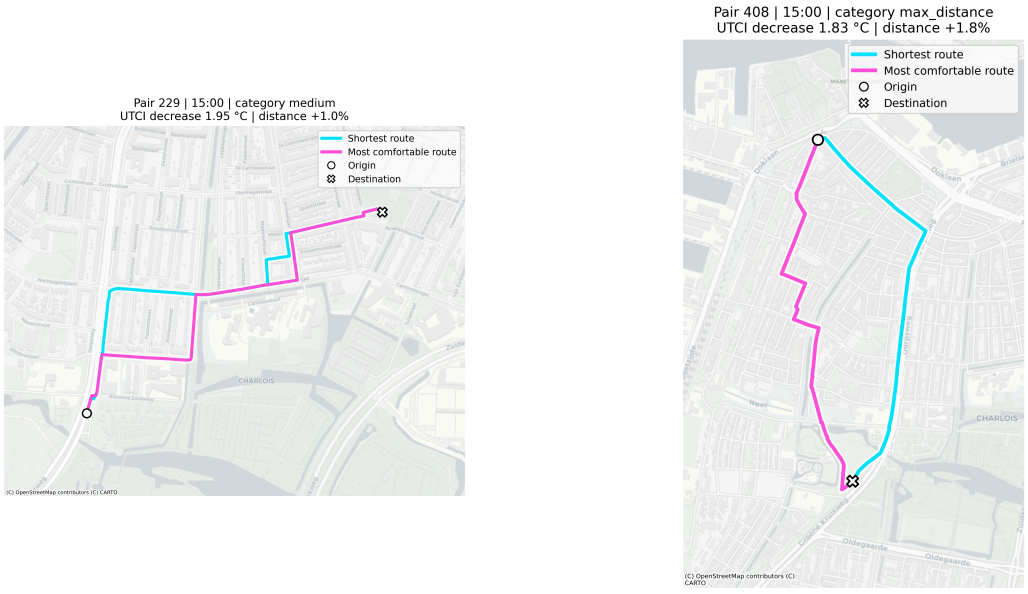

Figure 7.3: Shortest (cyan) and most thermally comfortable (magenta) path for OD pair 229 and 408,

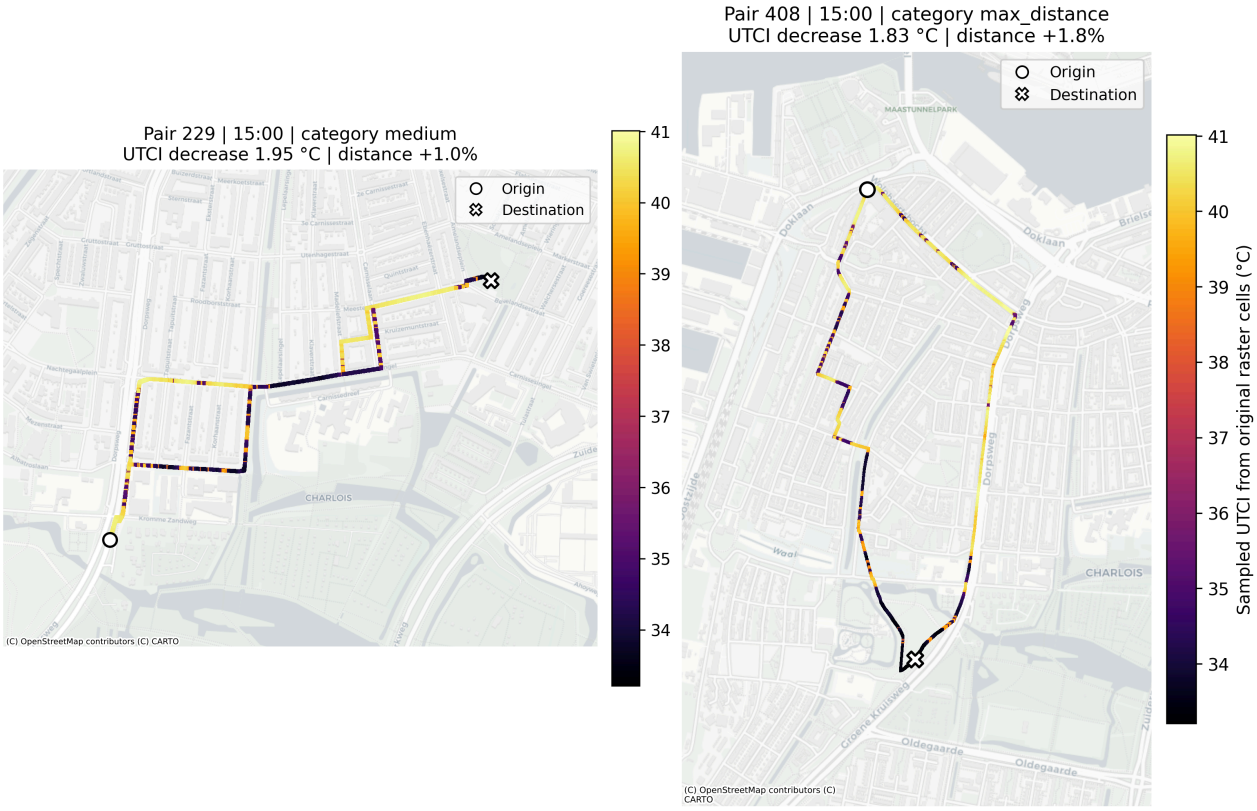

Figure 7.4: Sampled UTCI cells plotted on the shortest and most thermally comfortable path for OD pair 229 and 408.

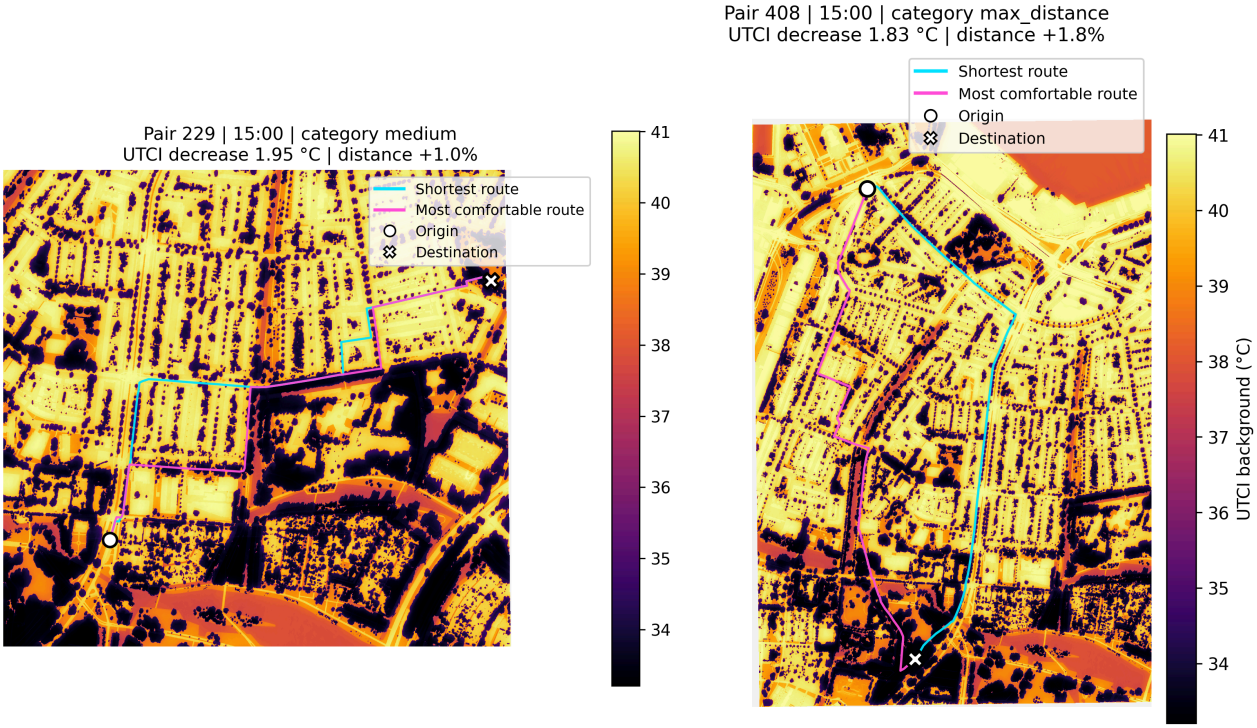

Figure 7.5: Shortest and most thermally comfortable path for OD pair 229 and 408, plotted on top of original UTCI map.

The paths shown in Figure 7.4 indicate which raster cells were sampled along each route. The cells sampled along the thermally conscious route generally have lower UTCI values than the corresponding alternatives along the shortest route. A closer inspection of the original UTCI map around the route in Figure 7.5[5] shows that the thermally conscious route mainly follows cooler areas associated with vegetation. These areas are visible on the UTCI map as rounded cool spots, which correspond to tree locations.

The increased presence of vegetation along the thermally conscious route is also visible in the satellite imagery shown in Figure 7.6. At the first deviation between the two routes for OD pair 229, the thermally conscious route follows `Roerdomplaan` and `Lepelaarsingel`, rather than the shortest alternative which sends you via `Dorpsweg` and `Wielewaalstraat`. The satellite imagery indicates that `Wielewaalstraat` contains relatively little vegetation, which is consistent with the higher UTCI values assigned to this segment. In contrast, `Roerdomplaan` is characterized by a greater presence of vegetation and is therefore selected by the routing algorithm as the thermally preferable alternative.

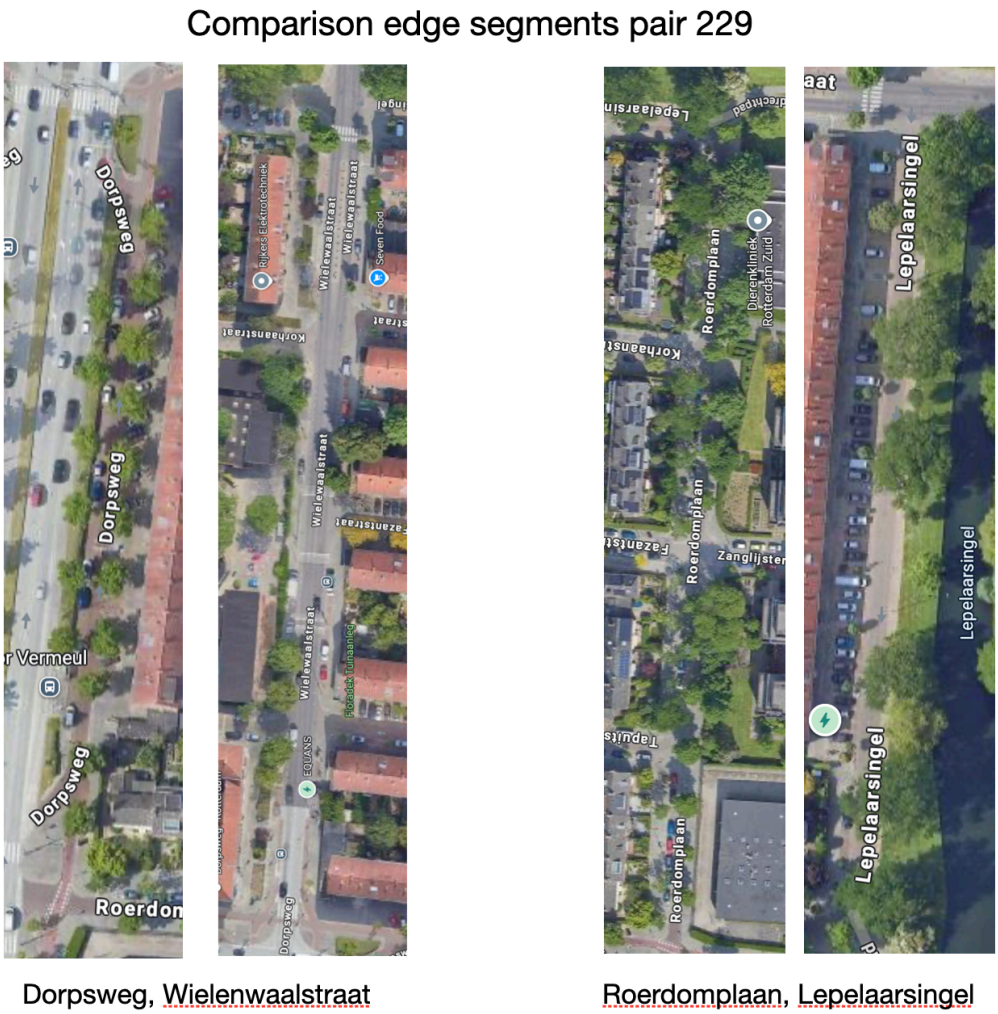

Figure 7.6: Satellite images of first deviant roads OD pair 229, roads in the shortest path (left) and roads in the most comfortable path (right).

[5] The figure legend scale is capped at the maximum value sampled along the route to increase contrast for route-level analysis. This was necessary because some disproportionately high UTCI values outside the route area skew the colour scale.

7.2.2.2. Structured

This analysis aims to show the general performance of the algorithm on the study area. The evaluation was based on a set of origin-destination pairs divided into five distance categories: `max_distance`, `medium_long`, `medium`, `short_mid`, and `short`. In total there were 5 distance categories and 100 OD pairs per category, all with different orientations and positions on vertices in the investigated area Figure 5.6. Each category contained 100 OD pairs with different positions and orientations across the pedestrian network. For every OD pair, the shortest route was calculated once and thermally comfortable routes were calculated for five hours: 08:00, 12:00, 15:00, 18:00, and 22:00. This resulted in 3000 route calculations in total.

Users can configure the relative importance of the routing weights. For these tests, the default settings were used, with the UTCI weight set to 1. Under this setting, the heat penalty is determined by the multiplier scheme described in Section 5 and Table 7, while consecutive exposure to hot edges is additionally penalized.

The results are evaluated using three main indicators. The first is route distance, which shows how much longer the comfortable route becomes compared with the shortest route. The second is average UTCI, which shows whether the comfortable route reduces thermal exposure. The third is route deviation, measured as the number of edges in the comfortable route that are not present in the shortest route. Together, these indicators show not only the trade-off between thermal comfort and additional walking distance, but also how different the routes are.

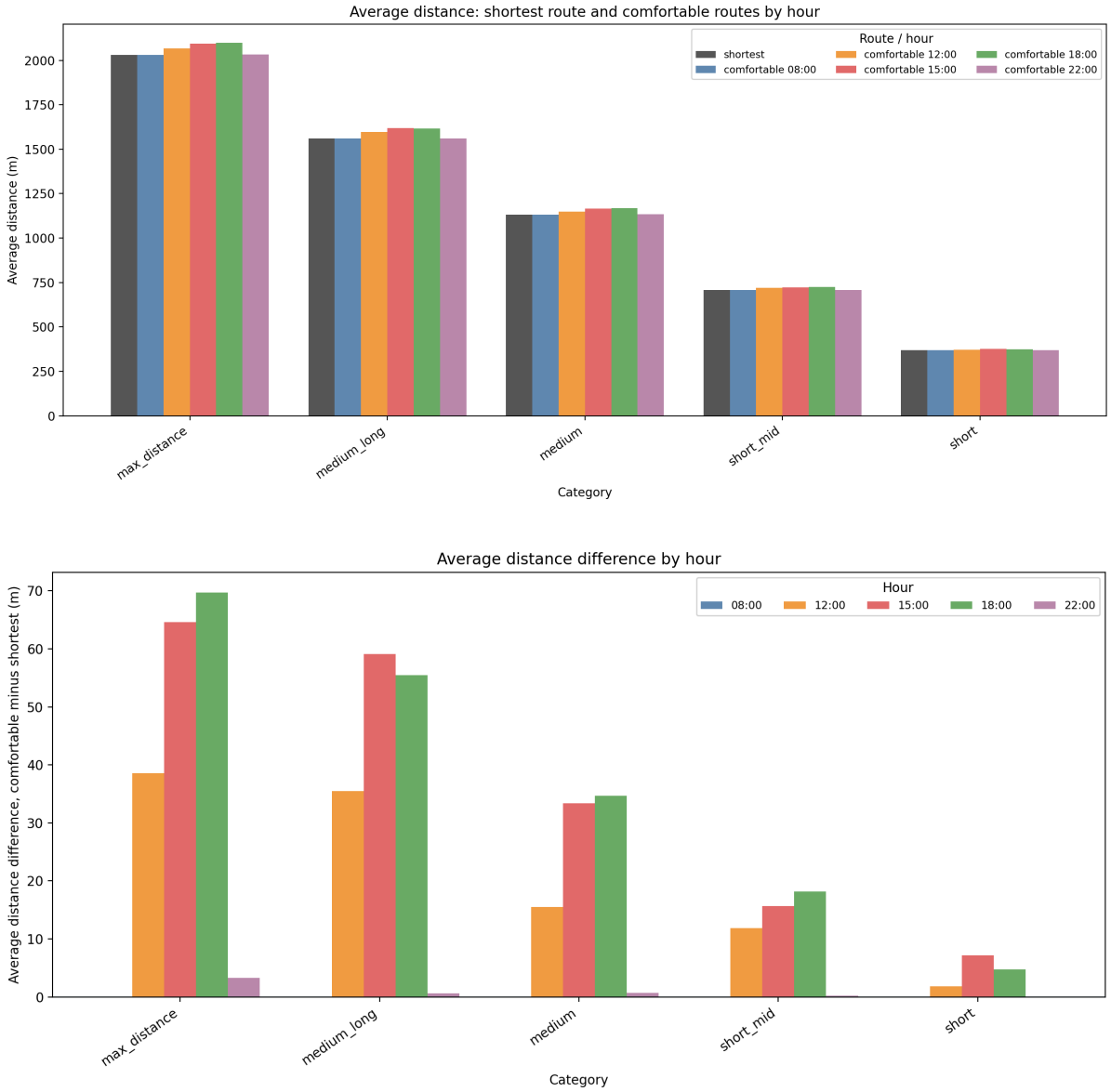

Figure 7.7: Absolute average distance of the routes (top), average difference in distance between shortest and most comfortable route (bottom)

The distance results in Figure 7.7 show that the comfortable routes are generally only longer than the shortest routes during the warmer hours. At 08:00 and 22:00, the average distance difference is close to zero for most categories, indicating that the comfortable route often remains identical or very similar to the shortest route. This is expected, because the algorithm is designed to penalize hot edges; if there are only a few hot edges at these times, there is little reason to select an alternative route. The largest distance increases occur at 15:00 and 18:00, especially for the longer distance categories, showing that the algorithm takes longer detours when avoiding heat becomes more relevant.

The average distance increase in the higher distance categories is about 70m which is the same as perceived distance increase caused by one extra degree in the UTCI [4]. The average distance increase for more comfortable routes is around 4% across categories which is a little lower than previous work [18, 35]. Indicating that there is potentially more room for deviation from the shortest path if additional variables are added to the algorithm.

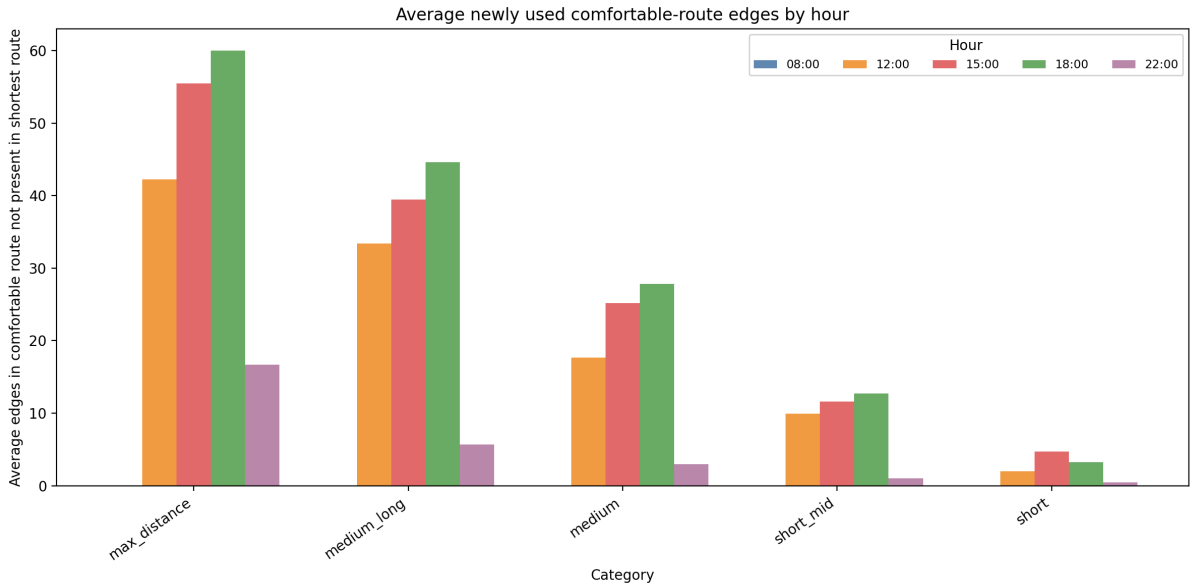

Figure 7.8: Amount of new segments (edges) in the comfortable route that are not in the shortest.

The number of different edges follows the same pattern as the distance increase Figure 7.8. Comfortable routes at 15:00 and 18:00 include more edges that are not part of the shortest route, particularly for the longer OD categories. This shows that the algorithm does not only increase route length slightly, but also selects partially different paths through the network when hot edges are present. At 08:00 and 22:00, the number of newly used edges is much lower, which again suggests that the shortest route often already avoids heat exposure, if it even is there, at these times.

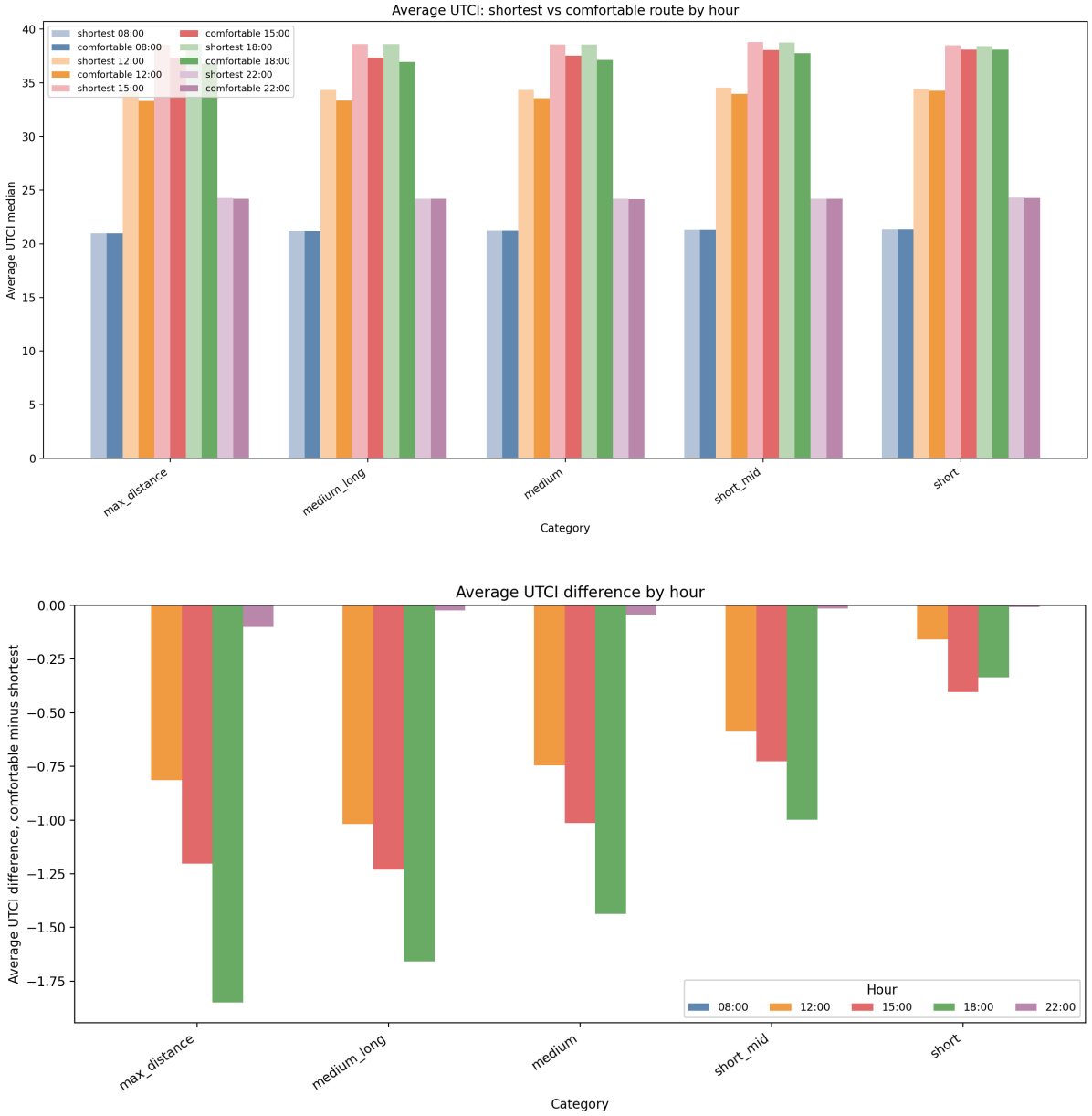

Figure 7.9: Absolute average UTCI of the routes both the UTCI of the shortest and the most comfortable route (top), average difference in UTCI between shortest and most comfortable route (bottom)

The UTCI results in Figure 7.9 show that the additional distance taken by the comfortable routes generally corresponds to a reduction in average UTCI. The route-level UTCI was calculated as an edge-length-weighted average, meaning that the UTCI value of each edge was weighted by its length before being averaged over the full route. As a result, longer edges have a larger influence on the route average than very short edges. The comfortable routes have a lower average UTCI than the shortest routes, especially during the warmer hours. The largest reductions occur at 15:00 and 18:00, while the differences at 08:00 and 22:00 are small or close to zero.

The reduction in UTCI at 12:00 is lowest of the 'hot times' measurements. This could be caused by the high position of the sun, which reduces the extent to which urban geometries such as buildings and vegetation cast shade. A similar effect was observed by Wen et al. (2025) at this time [39]. In addition, the limited number of edges classified into the highest UTCI categories, as shown by the lower average UTCI as compared to 15:00 and 18:00, may reduce the influence

of the heat-penalty factors. As a result, the routing algorithm has fewer strongly penalized edges to avoid and is therefore less likely to select a substantially different route. The reduction in UTCI is generally larger for the longer distance categories, where more alternative edges are available.

The reductions in UTCI are slightly lower than those reported in previous work [18, 35]. This may partly be explained by a relatively lower importance assigned to heat avoidance in the routing weights, which also limits the additional distance the algorithm is willing to introduce. In the study by Rußig and Bruns (2017), the average route length was approximately 2 km; since UTCI reductions become more pronounced as route length increases, this may partly account for the larger reductions reported in their results. The comparison with Ma et al. (2025) should also be interpreted cautiously, as their results were obtained in the different climatic context of Hong Kong, where thermal conditions may differ substantially.

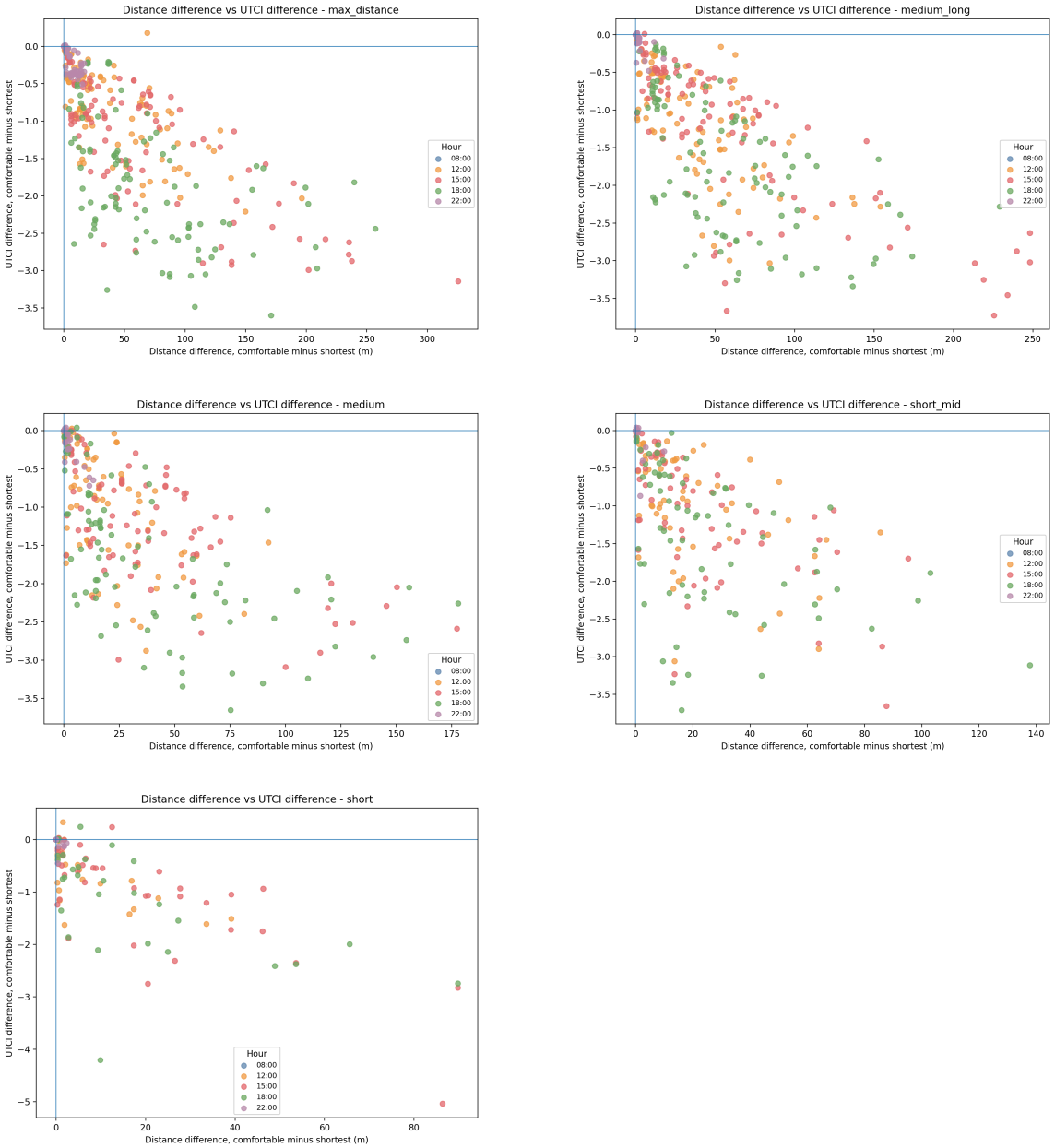

Figure 7.10: Distance (x-axis), UTCI difference (y-axis), between the shortest and most comfortable route

The scatterplots (Figure 7.10) show the trade-off between additional walking distance and UTCI reduction for individual OD pairs. The x-axis shows the distance difference between the comfortable and shortest route, while the y-axis shows the UTCI difference between the comfortable and shortest route. Because the UTCI difference is calculated as comfortable minus shortest, negative values indicate that the comfortable route has a lower average UTCI. Most points are located to the right of zero and below zero, meaning that the comfortable route is usually longer but cooler than the shortest route. The points for 15:00 and 18:00 show the clearest reductions, while 08:00 and 22:00 are clustered closer to zero. This again shows that the algorithm mainly changes the route during the hot parts of the day.

For a small number of OD pairs, the comfortable route has a higher average UTCI than the shortest route. This can occur because the algorithm strongly penalizes extremely hot edges. As a result, it may avoid one particularly hot edge by choosing a route with several slightly less hot edges. Although this behaviour reduces exposure to the most extreme heat conditions, it can increase the length-weighted average UTCI of the full route.

8. Discussion and conclusion

This chapter will first present the specific contributions, limitations and recommendations for future work of this thesis. Then it concludes by answering the research questions posed at the beginning of this thesis.

8.1. Contributions

- A fully open-source methodology for generating thermally conscious pedestrian routes, implemented in a locally deployable web-based tool for route interaction and visualisation.
- Insights into how implementation-level design choices affect the performance, scalability, and interpretability of urban microclimate-informed pedestrian routing tools.

8.2. Limitations

Climatic and geographic transferability: The results are based on a Rotterdam case-study area under one high-heat weather scenario. The weighting scheme and routing outcomes may therefore not transfer directly to cities with warmer, drier, or more extreme climates, or to other seasonal and meteorological conditions. Thus, the transferability of the proposed approach to other climatic contexts should be evaluated carefully before broader application.

Thermal-comfort indicator: The routing tool uses UTCI as a generalized indicator of thermal stress. UTCI is based on a predefined reference person and does not account for individual differences such as age or sex. As a result, the route planner cannot directly represent the thermal vulnerability of specific pedestrian groups. The main form of user customisation is limited to adjusting the user-defined weights.

Thermal-comfort validation: The validation done in this thesis (Section 7.2.2) shows that the tool can reduce modelled UTCI exposure compared with the shortest route, but it does not validate whether pedestrians would actually experience these routes as being more comfortable. No field measurements, human-subject testing, or perception-based validation were conducted.

SOLWEIG_GPU: The SOLWEIG_GPU implementation used in this thesis could not process negative height values, such as elevation values below sea level. Therefore, negative values in the DTM and DSM were set to zero before simulation. This preprocessing step may have introduced inaccuracies in the representation of the urban surface. Consequently, local variations in terrain and building height may not have been fully preserved in the thermal-comfort simulations.

Performance and scalability evaluation: The runtime results should be interpreted as indicative rather than as a general benchmark. They were measured on specific hardware and for a limited number of spatial cases. The thesis therefore demonstrates a scalable software structure, but not full large-scale operational scalability.

Pedestrian network representation: The pedestrian network used in this thesis is based on OpenStreetMap data and therefore inherits both the advantages and limitations of a crowd-sourced dataset. Tagging practices may vary between contributors, meaning that similar pedestrian features can be represented inconsistently or remain unmapped. In addition, many

road segments are represented as centre lines, while pedestrians for example often move along sidewalks. This may have introduced inaccuracies when sampling the UTCI raster, because the network geometry does not always correspond with the actual locations of pedestrian movement. The generated routes should therefore be interpreted as thermally conscious routes within the available digital network representation, rather than as exact representations of pedestrian-level exposure in the real world.

Automated abstraction of microclimate outputs: Urban microclimate modelling outputs are typically inspected and interpreted by domain experts as an end product for analysis, whereas in this thesis they form the basis for additional automated abstractions. This makes the workflow more vulnerable to modelling errors, preprocessing issues, or local anomalies that may remain unnoticed once the data are converted into routing weights. The risk is especially relevant for applications at scale, where results may be generated for larger areas or different contexts without detailed expert inspection of each intermediate output.

8.3. Future work

Scalability: The workflow can be applied to larger and more diverse study areas in order to evaluate its practical scalability. In this thesis, scalability was investigated mainly at the level of software structure, data formats, and component design. A next step would be to test whether the same workflow can be applied across different neighbourhoods, cities, climatic contexts, and weather scenarios without requiring substantial methodological redesign. This would make it possible to evaluate both the spatial transferability of the tool and the sensitivity of the routing outcomes to different thermal conditions.

Thermal comfort validation: The generated routes should be validated against actual pedestrian experience. This could be addressed for example through field measurements or feedback collected through the web-based interface. Such validation would be especially important for refining the weighting scheme, since the current routing costs are only based on modelled exposure rather than observed user preferences.

Person-specific and multi-variable routing: The current implementation uses UTCI as a generalized thermal-stress indicator and allows users to adjust the importance of thermal comfort through weighting. However, it does not directly model differences between pedestrian groups or individual vulnerability, even though vulnerable populations are at greater risk of heat-related health impacts. Future work could investigate the use of alternative thermal-comfort indicators, such as PET, if the required meteorological inputs are available. In addition, the routing framework could be extended with other environmental or experiential variables, such as greenery, shade, air quality, noise, or visual comfort. This would also allow further investigation of automatic or data-driven weighting schemes, rather than relying only on predefined or user-defined weights.

Performance: The most substantial performance bottlenecks occurred during data preparation and SOLWEIG execution. Future work could therefore explore cloud-native geospatial data access, which may reduce the need to merge complete source tiles when generating the input data before running the model. Alternative approaches to generating thermal-comfort maps, including machine-learning-based approximations [19], could also be investigated, particularly if they reduce computation time while preserving sufficient spatial accuracy for routing applications. Once the effect of spatial resolution on routing accuracy has been validated, lower-resolution simulations could also be explored as a performance optimisation strategy.

Pedestrian network: Future research could compare centre-line-based networks with sidewalk-aware networks, manually corrected pedestrian geometries, or datasets that include indoor and semi-indoor pedestrian paths. This would make it possible to quantify how network representation affects sampled UTCI values, route selection, and the estimated thermal benefit of the generated routes.

8.4. Research overview

8.4.1. Main research question

"**RQ:** *How can urban microclimate data generated by UMEP's SOLWEIG be transformed and integrated into a fully open-source, scalable pedestrian routing tool to support thermally conscious mobility decisions at the individual level?*"

In this thesis, SOLWEIG-derived UTCI outputs are operationalized by precomputing hourly raster outputs, sampling them onto a pedestrian network, and using the resulting edge-level attributes in a modified weighted-cost routing algorithm. The workflow that does this consists of several separate components, each with its own function, data requirements, and performance bottlenecks. As a result, there is no single data structure that is optimal for the entire workflow. Instead, the representation of the data should be adapted to the task of each component. This requires a modular software design in which related functionality is grouped within cohesive components, while dependencies between components remain limited. The open-source design is central to this workflow, because it allows the transformation from SOLWEIG-based microclimate modelling to route generation to remain reproducible, inspectable, and adaptable beyond the case-study implementation.

This modular design is reflected in the way spatial information changes form throughout the workflow. SOLWEIG outputs represent thermal conditions as gridded raster surfaces, which are appropriate for microclimate modelling but not directly usable by a graph-based routing algorithm. During network annotation, these raster values are sampled and aggregated into edge-level attributes. These attributes then become part of the routing cost, allowing thermal conditions to be incorporated into path calculation without requiring the routing algorithm to access the full raster output directly.

In this way, the tool supports thermally conscious mobility decisions by translating urban microclimate information into a route alternative for an individual trip. Rather than only providing a map of thermal conditions, the workflow makes this information actionable by using modelled heat exposure directly in route generation. The resulting route reflects a trade-off between distance and thermal comfort based on the selected routing settings, allowing the user to choose a route that is informed by local thermal conditions.

8.4.2. Operationalizing UMEP outputs

"**RQ1:** *What design strategies and data structures enable the effective use of UMEP outputs in a scalable pedestrian route planning application?*"

The first design strategy is to make the preparation of input data as automatic and reusable as possible. Because UMEP/SOLWEIG simulations require several spatial and meteorological inputs, the workflow should avoid manual, case-specific preparation steps where possible. This improves reproducibility and makes it easier to apply the same workflow to other study areas. However, the actual performance of running SOLWEIG remains dependent on the available hardware and the size and resolution of the input data. Therefore, the workflow can

be designed to support scalability, but exact computation times cannot be generalized without testing them in a specific computing environment.

A second important strategy is to transform the SOLWEIG outputs into a data structure that matches the needs of the routing application. Instead of using the full raster outputs directly during route calculation, the environmental information is sampled in advance and attached to the pedestrian network. In this way, the pedestrian network becomes the main data structure for routing as well as for storing relevant environmental attributes. This structure is also functionally extendable, because additional variables can be appended to the network edges and described in the metadata without changing the overall system design.

Lastly, to support this transformation, appropriate file formats for the use-case should be chosen. In this thesis, the SOLWEIG outputs were merged into a Zarr file, which supports efficient handling of large spatio-temporal raster data. The performance improvement this caused shows that these file format choices are not only storage decisions, but directly affect the performance of the workflow.

8.4.3. Routing algorithm

"**RQ2:** *How can established weighted-cost routing algorithms incorporate SOLWEIG outputs and be implemented and adapted to generate thermally conscious pedestrian routes?*"

The thermal attributes derived from SOLWEIG are attached directly to the pedestrian network and used by the cost function of the routing algorithm. Edges with less favourable thermal conditions receive a higher cost, allowing the algorithm to generate thermally conscious routes. For the algorithm to be applicable in a thermally conscious routing context, it needed to incorporate both dynamic thermal exposure and adjustable user preferences.

First, the algorithm needed to be dynamic, because the cost of the next edge can depend not only on its environmental attributes, but also on the current state of the route and the edges that have already been taken. Therefore, the shortest path algorithm needs to be state-aware. This was implemented by adapting Dijkstra's algorithm so that it stores costs for combinations of node and route state, rather than a single least-cost value per node. The route state represents accumulated heat exposure, allowing previous exposure to influence the cost of subsequent edges. Second, the routing should not depend only on researcher-defined weights. By allowing users to adjust the importance of environmental attributes, the tool can better account for different preferences and changing circumstances.

8.4.4. Scalability

"**RQ3:** *What does scalability mean for a pedestrian routing tool particularly regarding its potential for spatial and functional extension?*"

In the context of this thesis, scalability refers to the ability of the pedestrian routing tool to be extended beyond its initial case-study implementation without requiring fundamental redesign. This includes spatial scalability, meaning application to larger or different geographic areas, and functional scalability, meaning the ability to incorporate additional routing criteria as the definition of thermally comfortable pedestrian routing develops. Scalability is therefore not limited to computation time, but also concerns whether the tool's structure can support future adaptation.

For this type of workflow, scalability depends on three related implementation principles: performance, ease of use, and reusability through modular design. Performance is necessary because spatial extension increases the number of raster cells, network edges, and possible

route calculations, which can amplify existing bottlenecks in data preparation, network annotation, and routing. At the same time, performance should remain tied to the intended use case: components need to be fast enough to support practical use and interactive routing, but not optimized beyond what provides meaningful benefit. Ease of use is also central to scalability, because workflows that depend on extensive manual intervention become slower and more error-prone. Finally, reusability supports scalability by separating the workflow into components with clearly defined purposes and explicit data exchanges, allowing individual parts to be tested, replaced, or extended without redesigning the complete tool.

8.5. Conclusion

The broader implication of this thesis is that scalability in urban microclimate-informed routing cannot be separated from implementation. Data structures, file formats, software architecture, and network representations shape not only computational performance, but also the interpretability of the resulting routes. Treating these choices as methodological decisions rather than neutral technical details makes it possible to identify performance bottlenecks and address them through targeted design choices, without necessarily changing the underlying research objective. Scalability can also serve as a condition for broader adoption: if tools like this are to be applied beyond a single case-study area, they must remain efficient, reusable, and adaptable in different urban contexts.

This supports the aim introduced at the start of the thesis: contributing to the adaptive capacity of urban communities under increasingly hot weather conditions. The results show that existing urban infrastructure can be used more intelligently by accounting for modelled thermal conditions during route generation. Further validation with experienced thermal comfort is needed to refine the balance between additional distance and thermal benefit, but the workflow demonstrates that microclimate-informed routing can provide an actionable way for individuals to make more thermally conscious mobility decisions.

A Appendix A

Table only has insights from the emergent design process. If the first methodological approach did not encounter any major bottlenecks, it is not noted here but only in the methodology Section 5. Because the methodology was primarily informed by previous work, several emergent insights for one topic indicate aspects of the workflow that appear to have received less detailed attention in the literature consulted for this thesis.

A.1 Table emergent process

Insight	Component	Initial approach	Bottleneck or observation	Revised approach or implication	Broader lesson
I1	**Data preparation**	Use the SOLFD pipeline [23].	The pipeline was difficult to use because of how the code was distributed.	Distribute the SOLFD pipeline as a PyPI package.	A scalable pipeline must not only be technically functional, but also easy to install, run, and reproduce.
I2	**Pedestrian network**	Retrieve the pedestrian network from OpenStreetMap.	The resulting network consists mainly of centreline geometries, which assumes that pedestrians move along the centre of each mapped edge.	Treat this as a limitation.	Pedestrian network preparation is not a neutral input step, though it is currently the best we have. The geometric abstraction can introduce inaccuracies into the final routing results.
I3	**Pedestrian network**	Use existing automatic OSM filtering through abstraction tools such as OSMnx.	Automatic filtering can remove tags with sidewalk and footway information.	Load all available OSM information and apply project-specific filtering.	Software automations can introduce unnoticed data loss; the choice of data collection tool can affect the final result.
I4	**Network representation**	Use the standard OSMnx representation, where bidirectional movement is represented through duplicate directed edges.	Bidirectional pedestrian connections increased storage demand.	Use an undirected internal network representation.	Standard network representations should be questioned when their assumptions do not match the workflow.
I5	**SOLWEIG / UMEP execution**	Run SOLWEIG on demand for only the area directly needed by the route.	Input data collection from tiles and CHM generation were too slow for interactive, on-demand routing.	Run SOLWEIG pre-emptively and integrate the environmental outputs into the pedestrian network.	On-demand environmental simulation is only feasible when input-data retrieval and preprocessing are fast enough for interactive use.
I6	**SOLWEIG / UMEP execution**	Run SOLWEIG on external compute infrastructure when available.	SOLWEIG_GPU performance depends strongly on available GPU memory, tile size, and shared hardware conditions.	Test SOLWEIG_GPU on the available hardware.	SOLWEIG performance is highly hardware-dependent, so scalability should be evaluated in relation to the specific computing environment.
I7	**SOLWEIG / UMEP execution & network annotation**	Merge tiled SOLWEIG outputs into a multidimensional GeoTIFF.	The resulting files were large, and multidimensional lookup was inefficient during pedestrian-network annotation.	Store tiled environmental outputs in a Zarr structure.	File format choices should match the needs of the computation.

Insight	Component	Initial approach	Bottleneck or observation	Revised approach or implication	Broader lesson
I8	**Network annotation & routing prototype**	Use the OSMnx / NetworkX graph representation directly in routing.	Implementation is dependent OSMNX graph representation which couples the routing and network allocation by file format. Introduces performance bottlenecks for routing.	Use a shared data schema between components, while allowing the internal representation to change between use-cases.	Dependencies on static network representations can reduce performance, flexibility, and reusability.
I9	**Routing prototype & application**	Let the application also calculate routes.	Coupling the application and routing logic reduced reusability, and client-side route calculation can introduce performance constraints at scale.	Separate the routing engine from the application and communicate between them through an API.	The application should primarily visualize, while the routing logic is done in the backend.

B Declaration of AI/LLM usage

Help from ChatGPT was employed in this thesis for the following purposes: improving sentence structure and academic wording, feedback on written sections, searching for academic literature and coding. All the results, analyses and conclusions are my own, and I verified that the content is original and meets academic integrity standards.

C Reproducibility self-assessment

All input data necessary can be generated using the workflow developed in this thesis. Due to the file size of the urban microclimate data it was not possible to upload this data to GitHub. All code is available in this repository, since the code is divided over 3 repositories these are published together in an organisation. Complete instructions for running the code are included in the README's of the folders. I would rate the reproducibility of the code **Moderate/High**, since the UTCI.zarr is the only data that is missing to run the full workflow.

Bibliography

[1] R. Aghamolaei, M. M. Azizi, B. Aminzadeh, and J. O'Donnell, "A comprehensive review of outdoor thermal comfort in urban areas: Effective parameters and approaches," *Energy & Environment*, vol. 34, no. 6, pp. 2204–2227, Sep. 2023, doi: 10.1177/0958305X221116176.

[2] G. Aliyev and M. Nanni, "Exploiting Vehicular Data for Exposure-Aware Pedestrian Routing," in *2025 26th IEEE International Conference on Mobile Data Management (MDM)*, Irvine, CA, USA: IEEE, Jun. 2025, pp. 11–19. doi: 10.1109/MDM65600.2025.00022.

[3] G. Aliyev and M. Nanni, "Vehicle-Pedestrian Optimization Framework for Exposure-Aware Routing," *Mobile Networks and Applications*, Oct. 2025, doi: 10.1007/s11036-025-02459-4.

[4] R. Basu, N. Colaninno, A. Alhassan, and A. Sevtsuk, "Hot and bothered: Exploring the effect of heat on pedestrian route choice behavior and accessibility," *Cities*, vol. 155, p. 105435, Dec. 2024, doi: 10.1016/j.cities.2024.105435.

[5] G. Boeing, "Modeling and Analyzing Urban Networks and Amenities With OSMnx," *Geographical Analysis*, vol. 57, no. 4, pp. 567–577, 2025, doi: https://doi.org/10.1111/gean.70009.

[6] K. Błażejczyk, G. Jendritzky, P. Bröde, D. Fiala, G. Havenith, Y. Epstein, A. Psikuta, and B. Kampmann, "An Introduction to the Universal Thermal Climate Index (UTCI)," *Geographia Polonica*, vol. 86, no. 1, pp. 5–10, 2013, doi: 10.7163/GPol.2013.1.

[7] Delft High Performance Computing Centre (DHPC), "DelftBlue Supercomputer (Phase 2)." 2024.

[8] E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959, doi: 10.1007/BF01386390.

[9] Y. Du, T. Aoki, and N. Fujiwara, "A review of human mobility: Linking data, models, and real-world applications," *Journal of Computational Social Science*, vol. 8, no. 4, p. 90, Nov. 2025, doi: 10.1007/s42001-025-00414-7.

[10] K. Foshag, J. Fürle, C. Ludwig, J. Fallmann, S. Lautenbach, S. Rupp, P. Burst, M. Betsch, A. Zipf, and N. Aeschbach, "How to Assess the Needs of Vulnerable Population Groups Towards Heat-sensitive Routing?," *Erdkunde*, vol. 78, no. 1, pp. 1–34, 2024, doi: https://doi.org/10.3112/erdkunde.2024.01.01.

[11] GeeksforGeeks, "Adjacency List Representation." Jul. 23, 2025.

[12] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design Science in Information Systems Research," *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004, Accessed: Apr. 21, 2026. [Online]. Available: http://www.jstor.org/stable/25148625

[13] IPCC, "Summary for Policymakers," in *Climate Change 2022: Impacts, Adaptation, and Vulnerability. Contribution of Working Group II to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*, H. O. Pörtner, D. C. Roberts, M. Tignor, E. S. Poloczanska, K. Mintenbeck, A. Alegría, M. Craig, S. Langsdorf, S. Löschke, V. Möller, A. Okem, and B. Rama, Eds., Cambridge, UK: Cambridge University Press, 2022, p. In Press.

[14] H. G. Kamath, N. Sudharsan, M. Singh, N. Wallenberg, F. Lindberg, and D. Niyogi, "SOLWEIG-GPU: GPU-Accelerated Thermal Comfort Modeling Framework for Urban Digital Twins," *Journal of Open Source Software*, vol. 11, no. 118, p. 9535, Feb. 2026, doi: 10.21105/joss.09535.

[15] S. Khalid and B. Hussain, "Software reusability and component-based architecture," *Multidisciplinary Research in Computing Information Systems*, vol. 5, no. 5, pp. 497–505, May 2025, doi: 10.71465/mrcis108.

[16] Koninklijk Nederlands Meteorologisch Instituut (KNMI), "Daggegevens van het weer in Nederland: station 344 Rotterdam Airport." [Online]. Available: https://www.daggegevens.knmi.nl/klimatologie/dag?date=2022-07-19&station=344

[17] F. Lindberg *et al.*, "Urban Multi-scale Environmental Predictor (UMEP): An integrated tool for city-based climate services," *Environmental Modelling & Software*, vol. 99, pp. 70–87, Oct. 2017, doi: 10.1016/j.envsoft.2017.09.020.

[18] X. Ma, T. Zeng, R. De Dear, Y. Xie, C. Yuan, and S. Lu, "Active route choice to minimize pedestrian thermal discomfort in a high-density subtropical city," *Sustainable Cities and Society*, vol. 131, p. 106697, Sep. 2025, doi: 10.1016/j.scs.2025.106697.

[19] X. Ma, T. Zeng, M. Zhang, P. Zeng, B. Lin, and S. Lu, "Street microclimate prediction based on Transformer model and street view image in high-density urban areas," *Building and Environment*, vol. 269, p. 112490, Feb. 2025, doi: 10.1016/j.buildenv.2024.112490.

[20] E. Manley, G. Filomena, and P. Mavros, "A spatial model of cognitive distance in cities," *International Journal of Geographical Information Science*, vol. 35, no. 11, pp. 2316–2338, Nov. 2021, doi: 10.1080/13658816.2021.1887488.

[21] B. Mehboob, C. Y. Chong, S. P. Lee, and J. M. Y. Lim, "Reusability affecting factors and software metrics for reusability: A systematic literature review," *Software: Practice and Experience*, vol. 51, no. 6, pp. 1416–1458, Jun. 2021, doi: 10.1002/spe.2961.

[22] G. Mihalakakou, J. A. Paravantis, P. Nikolaou, S. Malefaki, A. Romeos, A. Fotiadi, P. N. Georgiou, and A. Giannadakis, "Urban Canyon Geometry and Green Infrastructure: A Review of Strategies for Enhancing Thermal Comfort and Microclimate," *Sustainability*, vol. 18, no. 9, 2026, doi: 10.3390/su18094335.

[23] J. Monahan, "Cool by design. SOLFD: Extending SOLWEIG for Urban Design Decision Making on Outdoor Thermal Comfort," Master's thesis, Delft, 2025.

[24] J. Monahan, "SOLWEIG_SOLFD." GitHub, 2025.

[25] G. Mora-Navarro, C. Femenia-Ribera, J. Martinez-Llario, and E. Antequera-Terroso, "Optimising urban routes as a factor to favour sustainable school transport," *Journal of Transport Geography*, vol. 72, pp. 211–217, Oct. 2018, doi: 10.1016/j.jtrangeo.2018.09.001.

[26] D. L. Morgan, "Emergent design," *The SAGE Encyclopedia of Qualitative Research Methods*, no. 0, p. , 2008, doi: https://doi.org/10.4135/9781412963909.n128.

[27] T. Novack, Z. Wang, and A. Zipf, "A System for Generating Customized Pleasant Pedestrian Routes Based on OpenStreetMap Data," *Sensors*, vol. 18, no. 11, pp. 37–94, Nov. 2018, doi: 10.3390/s18113794.

[28] T. R. Oke, G. Mills, A. Christen, and J. A. Voogt, "Climates of Humans," in *Urban Climates*, Cambridge University Press, 2017, pp. 385–407.

[29] T. R. Oke, G. Mills, A. Christen, and J. A. Voogt, "Glossary and Acronyms," in *Urban Climates*, Cambridge University Press, 2017, pp. 469–485.

[30] OpenStreetMap contributors, "Planet dump retrieved from https://planet.osm.org ." 2017.

[31] L. Pappalardo, E. Manley, V. Sekara, and L. Alessandretti, "Future directions in human mobility science," *Nature Computational Science*, vol. 3, no. 7, pp. 588–600, Jul. 2023, doi: 10.1038/s43588-023-00469-4.

[32] H.-O. Pörtner *et al.*, *Climate Change 2022: Impacts, Adaptation and Vulnerability*. in Technical Summary. Cambridge, UK and New York, USA: Cambridge University Press, 2022, pp. 37–118.

[33] J. J. Van Rensburg and R. Goede, "A Model for Improving Knowledge Generation in Design Science Research through Reflective Practice," *Electronic Journal of Business Research Methods*, vol. 17, no. 4, Dec. 2019, doi: 10.34190/JBRM.17.4.001.

[34] P. Runeson, E. Engström, and M.-A. Storey, "The Design Science Paradigm as a Frame for Empirical Software Engineering," *Contemporary Empirical Methods in Software Engineering*. Springer International Publishing, Cham, pp. 127–147, 2020. doi: 10.1007/978-3-030-32489-6_5.

[35] J. Rußig and J. Bruns, "Reducing Individual Heat Stress through Path Planning," *GI_Forum*, vol. 1, pp. 327–340, 2017, doi: 10.1553/giscience2017_01_s327.

[36] A. Salazar Miranda, Z. Fan, F. Duarte, and C. Ratti, "Desirable streets: Using deviations in pedestrian trajectories to measure the value of the built environment," *Computers, Environment and Urban Systems*, vol. 86, p. 101563, Mar. 2021, doi: 10.1016/j.compenvurbsys.2020.101563.

[37] N. Sudharsan, HarshKamath, and M. Singh, "SOLWEIG_GPU." GitHub, 2025.

[38] R. Trujillo and M. Solar, "Incremental Data Cube Architecture for Sentinel-2 Time Series: Multi-Cube Approaches to Dynamic Baseline Construction," *Remote Sensing*, vol. 18, no. 2, 2026, doi: 10.3390/rs18020260.

[39] J. Wen, D. A. Abuhani, M. Mazzarello, F. Duarte, L. Norford, R. Xu, N. H. Wong, and C. Ratti, "Walking smart in the heat: A dynamic shade-oriented pathfinding approach to enhance pedestrian comfort in arid cities," *Computers, Environment and Urban Systems*, vol. 122, p. 102337, Dec. 2025, doi: 10.1016/j.compenvurbsys.2025.102337.

[40] C. Wohlin and P. Runeson, "Guiding the selection of research methodology in industry–academia collaboration in software engineering," *Information and Software Technology*, vol. 140, p. 106678, 2021, doi: https://doi.org/10.1016/j.infsof.2021.106678.

[41] D. Zendeli, N. Colaninno, D. Maiullari, M. Van Esch, A. Van Timmeren, G. Marconi, R. Bonora, and E. Morello, "From heatwaves to 'healthwaves': A spatial study on the impact of urban heat on cardiovascular and respiratory emergency calls in the city of Milan," *Sustainable Cities and Society*, vol. 124, p. 106181, Apr. 2025, doi: 10.1016/j.scs.2025.106181.

[42] H. Zhao, L. Zhao, Y. Zhai, L. Jin, Q. Meng, J. Yan, R. Wu, and R. D. Brown, "The impact of dynamic thermal experiences on pedestrian thermal comfort: A whole-trip perspective from laboratory studies," *Building and Environment*, vol. 258, p. 111599, 2024, doi: 10.1016/j.buildenv.2024.111599.

This document was typeset using Typst, and uses the template freely available at https://github.com/tudelft3d/msc_geomatics_thesis_typst.