Diplomarbeit

URBAN INFORMATION MODELLING

Keywords: Architectural design, urban information modelling, urban analysis, location based services

ausgeführt zum Zwecke der Erlangung des Akademisches Grades eines Diplom-Ingenieurs unter der Leitung

Ao.Univ.Prof.Dipl.-Arch.Dr.phil Georg Suter I E259 I Institute of Architecture Sciences

eingereicht an der Technischen Universität Wien

Fakultät für Architektur und Raumplanung

Boy d’Hont I 1428406

Wien, am 22. September 2017

ABSTRACT

This master thesis develops a software tool that aids architects in the process of urban analysis. The tool complements an urban 3D model with data from the Google Maps database, generating an "urban information model" in the process. The topics of urban analysis reach from the function of the building and time of building activity to the location of public transportation stations and density of services in the urban area. The generated charts and graphs assist the architect in the formulation of a hypotheses about the unknown urban area, making the software a practical tool for architectural competition design. First, the requirements for such an urban analysis toolare defined. A conceptual system design lays out the foundation for a software tool that can meet these requirements. The elaboration of these assets leads to the implementation of the urban analysis tool concept. A case study in Vienna and London test the capabilities of this prototype. Both case studies show that the software tool can give the architect a good overview of several cultural aspects in the urban area.

Die Diplomarbeit entwickelt ein Software Tool das Architekten unterstützt im Prozess der Stadtanalyse. Das Tool ergänzt ein städtebauliches 3D Model mit Daten aus den Google Maps Database und generiert ein "Urban Information Model" während das Prozess. Die Themen der Stadtanalyse greifen von Gebäudefunktionen und Gebäudeaktivitätszeit bis die Orte des Öffentliches Verkehr und Dienstintensität im Bereich. Die generierte Karte und Diagramme unterstützen dem Architekt in der Entwicklung des Hypothese betreffs ein unbekanntes Stadtumfeld und bildet damit ein wertvolles Tool für Architekturwettbewerbe. In der erste Schritt sind die Bedingungen für das Tool definiert. Ein konzeptuelles System Design gestaltet die Fundierung des Software Tools die die Bedingungen erfühlen können. Eine Ausarbeitung der Aspekte leitet nach eine Implementierung des Stadtanalyse Tool Konzept. Ein Case Study in Wien und London prüfen das Potential des Prototyps. Beide Case Studies zeigen dass das Tool Architekten ein gutes Überblick der diversen kulturellen Stadtaspekten bieten kann.

GLOSSARY

API	Application Programming Interface. A set of clearly defined methods of communication between various software components
Building Block Data (BL-D)	The object within the Urban Analysis Tool that contains all the information about the building blocks. Part of the Urban Information Model
Data Flow Model	A graphical representation of the flow of data through an information system
Development Sustainability Processing (DSP)	The process of defining the most sustainable development for a given site
GIS	Geographic Information System. A system to handle and present spatial or geographic data
Global coordinate	A coordinate in the real world, defined in latitude and longitude
Global Service Data	Service Data objects that have not been assigned to a Building Block Data object and are therefore stored in the top layer of the Urban Information Model
Local coordinate	A coordinate in the virtual axis system, defined in an X and Y value

Local Service Data

Service Data objects that have been assigned to a Building Block Data object and are therefore stored in the top layer of the respective Building Block Data object

LOD

Level of detail. Used to specify the detail level of a 3D model

Parsing

The process of analysing a string of symbols to extract a specific part of the string

Service Data (S-D)

The object within the Urban Analysis Tool that contains all the information about the public available services in the given area. Part of the Urban Information Model

Site Selection

The process of testing the level of fitness of multiple site for a given development plan

Transportation Network Data (TN-D)

The object within the Urban Analysis Tool that contains all the information about the transportation facilities in the given area. Part of the Urban Information Model

Type tag

Tags that the Google Maps API uses to specify the function of the object

Urban Analysis Tool

The product of this thesis. A tool that assists architects and designers in gaining more understanding about a given urban environment

Urban Information Model (UIM)

The object within the Urban Analysis Tool that contains all the information about the urban subject

TABLE OF CONTENTS

Abstract 3

Glossary 4

1. Problem definition 8

2. Related Work

2.1 The design work-flow 10

2.2 Types of site analysis 12

2.3 Data visualization 14

2.4 Existing urban simulation systems 16

2.5 Urban 3D data 20

2.6 GIS and its disadvantages 22

3. Requirements

3.1 Work-flow 24

3.2 3D data retrieval 26

3.3 Data analysis 26

3.4 Data visualization 26

3.5 Exporting data 26

4. System Design

4.1 System architecture 28

4.2 Conceptual work-flow 30

4.3 Input data 30

4.4 Static analysis functions 32

4.5 Dynamic analysis functions 32

4.6 Combining data 32

4.7 Data visualization 34

4.8 Exporting data 34

5. Implementation

5.1.1 Technical system architecture 36

5.1.2 Run-time environment 40

5.1.3 External libraries used 40

5.2 Technical work-flow 42

5.3 3D preparation, import and limitations 42

5.4 Analysis functions

5.4.1 Google API requests 44

5.4.2 Geocoding 48

5.4.3 Function retrieval 48

5.4.4 Public transportation 50

5.5 Combining data 50

5.6 Data visualization 52

5.7 Exporting data 52

6. Case Studies

6.1 Rathausstraße 1, Vienna, AT

6.1.1 General project information 54

6.1.2 Pre-existing 3D data 56

6.1.3 Results 58

6.2 Charterhouse Square, London, UK

6.2.1 General project information 66

6.2.2 Pre-existing 3D data 68

6.2.3 Results 70

6.3 Comparing the case studies 78

7. Discussion

7.1 Reflection on final product 82

7.2 Future development and improvement 82

7.3 Possible applications 83

References

Bibiolography 84

Figures 85

A. Code snippets

B. Diagrams and graphs

1.
PROBLEM DEFINITION

The digital revolution of the 20th century brought architects new tools to lighten the burden of their complex jobs. Big improvements in architectural drafting came with the development of digital CAD tools and, the more recent, uprising of Building Information Modelling. Nevertheless, these new instruments focus at the later stages of the architectural design process. A proper digital tool to guide the architect in the early stages of the design work-flow, the labour-intensive phase of collecting urban data and performing urban analysis, still lacks.

This thesis focuses at the development of an urban analysis tool that could decrease the required time for the retrieval of this urban information and the analysis. Secondly, it offers additional insight in high-complex urban tissue, in which design-influencing urban patterns come to light with time-consuming analysis.

The architectural competition resembles a scenario in the architectural practice where this tool could have a strong impact. This context binds the architect to a tight deadline, yet the project should hold a high level of architectural quality to qualify for a prize winning status. Currently, the easiest accessible category of information is the geometric environmental data, in the form of 2D plans or 3D models. The architect can often retrieve this data from the competition’s attachments. The city hall, archives or online commercial sources form alternative sources for this type of data. The retrieval of other types of data could prove more challenging. For example, information about important functions near the building site can only be obtained with a visit to the building site itself or with the use public sources like Google Maps. This data, among other types of information (traffic, monumental state), is accessible over commercial sources, though this data retrieval process may require a high budget or the necessary information lacks. Manual simulation of the scenario could provide an option in this situation, but this reality-approximating process requires lots of expertise and computing resources. In an ideal situation, the largest portion of the available design time should be spend on the creative process: the process of concept development, exploration and evaluation.

This thesis proposes a tool that focusses at the aspects of urban analysis that are relevant for architectural competitions, as this domain benefits from rapid urban insight. The range of data varies from the architectural geometric aspects, landscape properties and climatic circumstances to cultural heritage and legal restrictions. Automating the data retrieval will make the data more accurate and synchronised to real-world developments. Each of these raw data sets could offer value to the design process, but the interpretation of this data appears more important for competition design. If the architectural design team identifies complex, scenario-specific urban patterns early in the design process, it will provide a well-tailored end product that stands out from the competition. Therefore, the tool should combine the data in an, preferably graphical, overview that the architect can retrieve from and use in combination with mainstream digital drafting tools as CAD-, BIM- and/or 3D drawing software.

To summarise, the project aims to eliminate mundane analysis tasks in the early stages of the design process. To speed up the architectural design work-flow and grant more space for the core asset of architecture: creating.

2. RELATED WORK

2.1 The design work-flow

As shown in the figure on the right, the architectural design work-flow consist of three main design cycles: assignment research, conceptual design and detailed design (Batty, Dodge, Jang, Smith, 1998, p. 7). Assignment research offers insight in the newly presented scenario and problem. The phases of the assignment research are the data collection, in which the designer retrieves possible useful information, data analysis, in which the designer aims to find patterns in the found data, and problem definition, the conclusion drawn from of the found data patterns. The iterative nature of assignment research forces the architect to perform the design cycle multiple times for a neater defined problem definition. The conceptual design phase translates the problem definition to a physical design. In this phase, the user has to state design objectives, analyse different design solutions and implement these solution in new generated plans. The conceptual design phase also follows the iterative design work-flow, as the quality of the generated plans improve after multiple walks through the design cycle. The last phase is the detailed design. This phase has lots of similarities with the conceptual phase, but makes use of more specific real-world information to test the quality of the design. The final product of the design cycle is an architectural design that is ready for realisation.

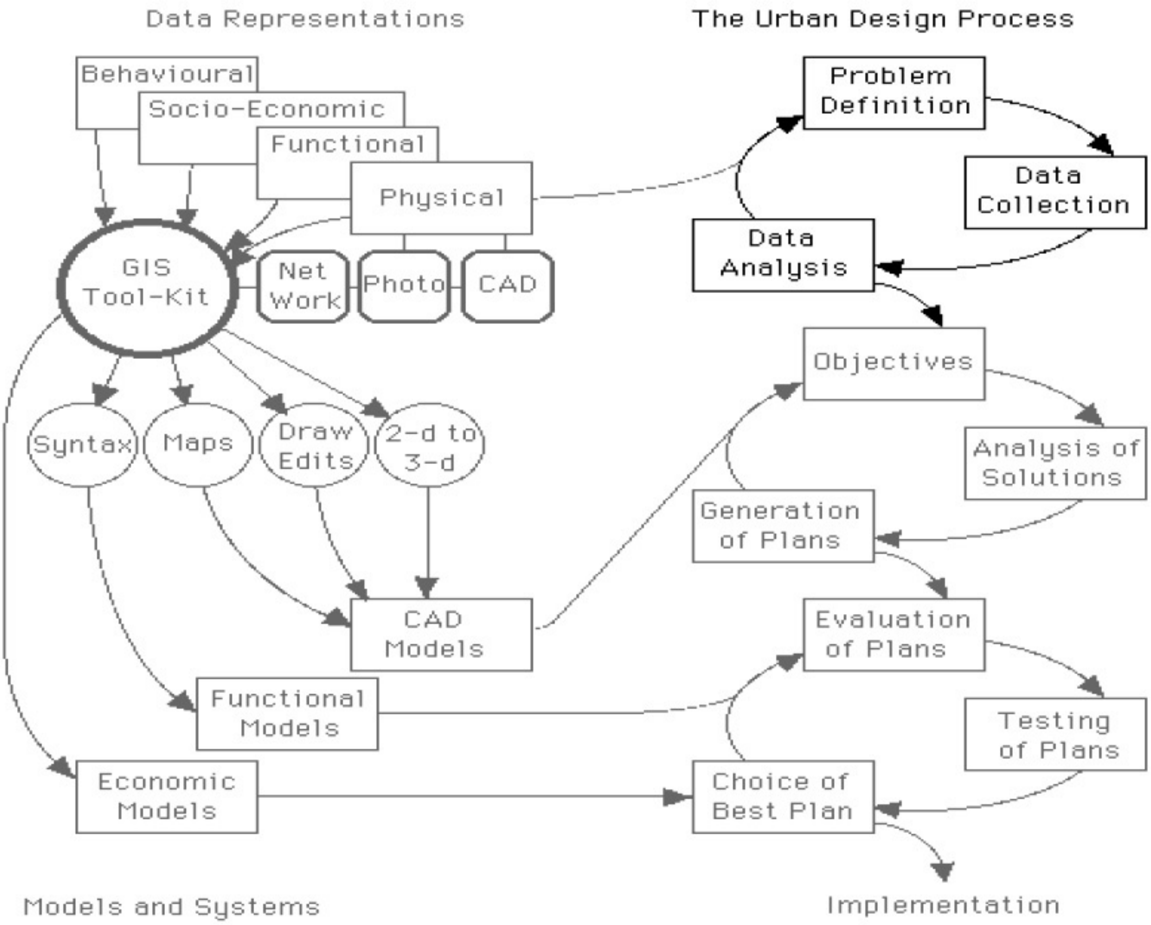


Figure 1: A visualization of the design process (edited copy of original) [1]

2.2 Types of site analysis

According to Kevin Lynch and Gary Hack (Lynch, Hack, 1974, p. 174), site establishment can be achieved using one of two methods: Development Sustainability Processing (DSP) and site selection. DSP focuses around one fixed site and aims to find the best possible use and development for that area. Site selection works the other way around: it starts with a fixed use and development plan and compares multiple sites for their suitability and compatibility with that program. Both types of site establishment require the same analysis, though the more accessible DSP method only requires the information for one site.

K. Lynch and G. Hack also stated a broad list of base analysis topics, possibly interesting for the architectural design process. They divide the analysis topics in three main categories: natural factors, cultural factors and aesthetic factors. Natural factors describe assets that do not belong to the build environment, like geomorphology, geology, vegetation and climate. Cultural factors describe social and economic aspects in the field of the build environment that can be defined in absolute data. This category contains analysis regarding traffic, current land use, social-economic factors and historic aspects. It also explores constraints for the future like legal building restrictions. Aesthetic factors characterize definitions that are usually difficult to capture in absolute data, like spatial qualities and points of orientation.

The retrieval of urban data succeeds with data sets (set data) or mathematical interpretation (interpreted data). Set data has no direct dependency at earlier obtained data. Interpreted data generation process has full dependency at previous data, as it does not retrieve new information from external sources on its own. The production of interpreted data transpires with the assistance of static formulas or spacial simulations.

Data interpretation with static formulas can describe concrete phenomena accurately. Landslide risk estimation, slope size and inclination computation, floor-area ratio calculation and grain size mapping illustrate methods that rely heavily on static formulas. The application of static formulas reach wider than concrete definitions, but also fit the characterisation process of abstract variables. For example, a case study of the design process of the high-tech campus of Shenzen describes the use of a mathematical analysis of very spiritual definitions. In this particular design process, a map showed favourable “Feng-Shui” zones (Fang, Xie,2008, p. 109), allowing the architects to base the urban design around religious beneficial areas. An addition of more well-being analysis, like safety perception, degree of disorientation and natural meeting points belong to the possibilities of this calculation method. Static interpretation only require little computing power, making it useful tool large quantities or complex calculations. Furthermore, the method creates the same output with the same input data, increasing the degree of result consistency.

Examples of calculations that mimic real world scenarios with spatial simulations are shadow studies, wind flow calculations, traffic density and pedestrian movement prediction and view axis definition. Spatial simulation often make use of space syntax, agent systems, cellular automata, ray-based systems or a hybrid of these multiple calculation methods. In contrast to static interpretation methods, simulations often need a fair amount of computing power. Therefore, simulation systems are often simplified to reduce time and resources that are required to complete the simulation. Finding the right balance between simulation simplification and data accuracy forms and important part of the simulation process. Simulation with the same input can result in different outcomes, as these methods often combine a high number of variables.

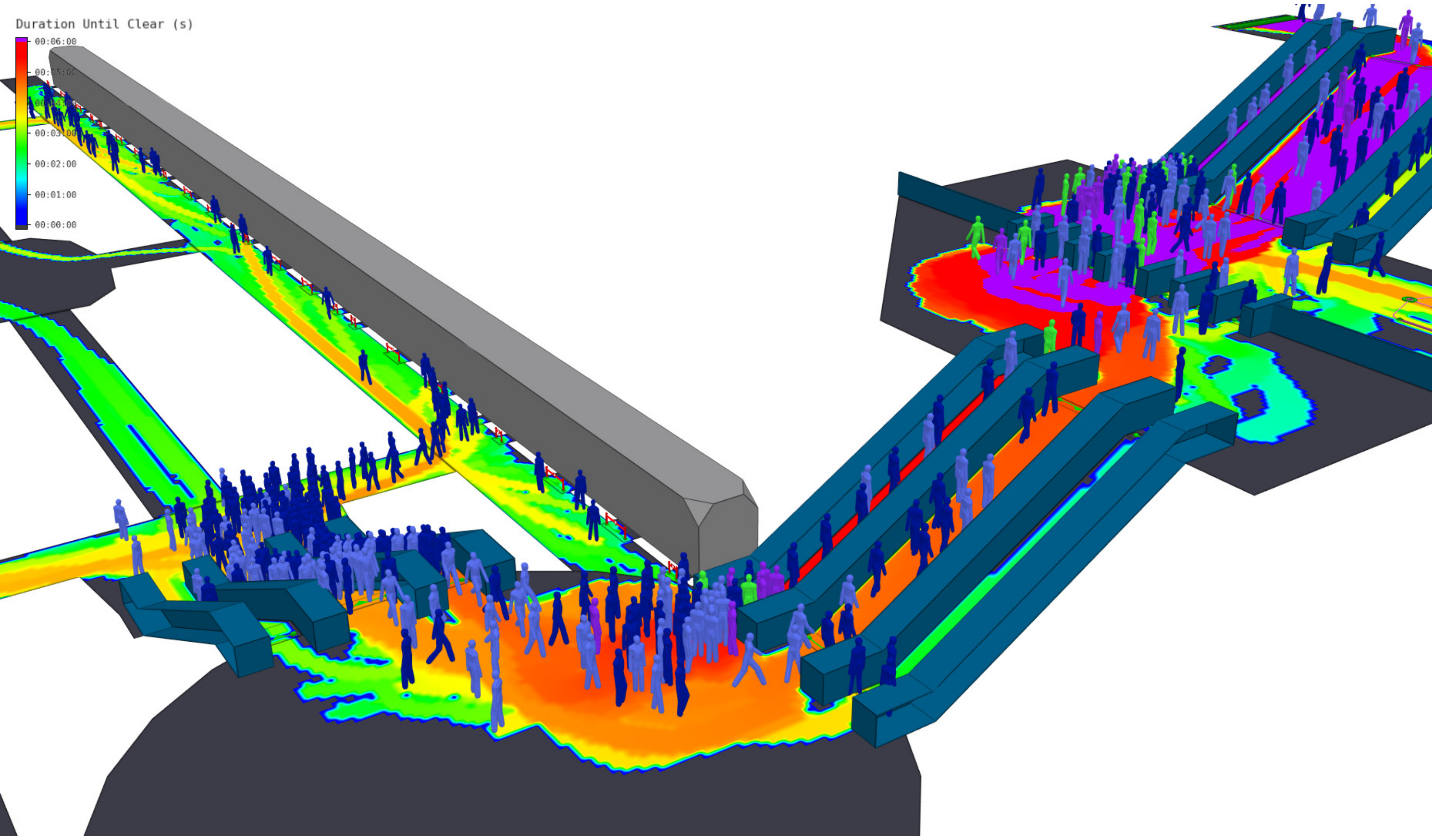


Figure 2: Agent-based fire escape simulation [2]

2.3 Data visualization

The helpful tool of data visualisation grants overview over large quantities of data. There are three goals for data visualization: exploration, conformation and presentation. Explorative data visualization starts without a clear hypothesis about the data that needs to be analysed. The visualization of the data should help with the development of a hypotheses about the data. This visualisation process should be interactive, as the user should have the possibility to search for data structures and patterns with minimal restrictions or direction. Conformative data visualization starts with a hypotheses about the data and only seeks to find proof to back up the hypotheses. This means the goal-oriented method offers a visualization of the data that allows the conformation or rejection of the hypotheses. Presentation visualisations only seek to display found facts and make them easy to understand to third parties. This visualisation type should focus at the visual quality of the visualisation, rather than offering the user the possibility to find new information or patterns.

Data visualization techniques can be categorised in several groups, each offering their own advantages and disadvantages (Keim, 1997 page 4). The classified data visualization techniques are geometric visualisations, icon-based visualisations, pixel oriented techniques, hierarchical techniques, graph-based techniques and hybrid visualisations. Geometric visualisations manifest geometric transformations and projections of the data. Scatter-plot matrices and landscape visualisations fall under this category. This method fits well for data with a low amount of variables that require comparison.

Icon-based visualisation adds a layer of symbols so the user can clearly distinguish different groups within the visualization. Colour icons and stick figures can serve as symbol layers. Pixel-based techniques represent each attribute value by one coloured pixel. These techniques prove useful when the user needs to find patterns in large quantities of data. Colour icon visualisation works especially well with categorical data.

Hierarchical techniques make use of subspaces. Examples of hierarchical techniques are dimensional stacking and tree-maps. Hierarchical techniques make it easy for the user to search for global and local patterns at the same time.

Graph-based techniques also work well in situation where few variables require comparison. The techniques differ in their foundations: geometric visualization bases on fields, graphs utilize line figures. Graphs have 2D and 3D variation and can contain branches. Hybrid visualization combines multiple categories of visualization techniques.

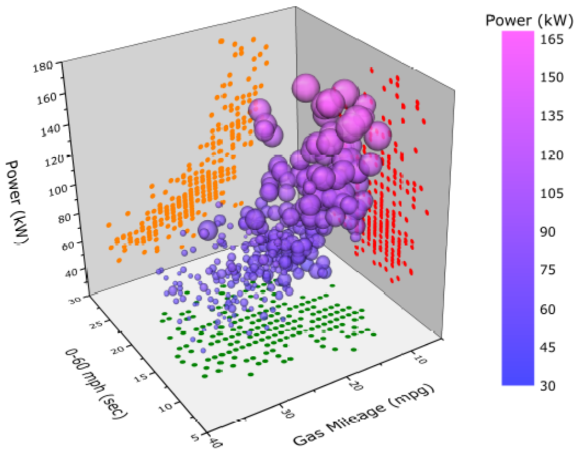
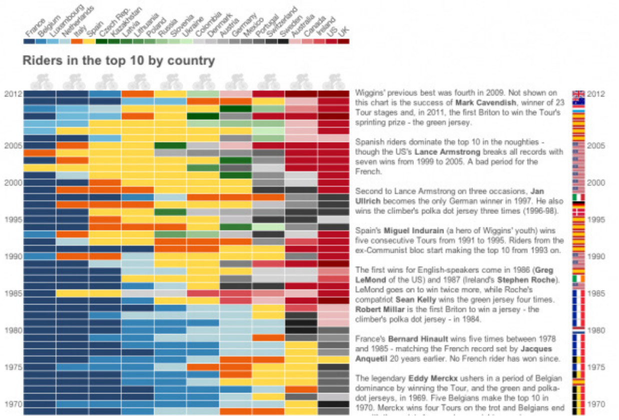


Figure 3: 3D scatter plot visualisation for three variables [3]



Wiggins' previous best was fourth in 2009. Not shown on this chart is the success of **Mark Cavendish**, winner of 23 Tour stages and, in 2011, the first Briton to win the Tour's sprinting prize - the green jersey.

Spanish riders dominate the top 10 in the noughties - though the US's **Lance Armstrong** breaks all records with seven wins from 1999 to 2005. A bad period for the French.

Second to Lance Armstrong on three occasions, **Jan Ullrich** becomes the only German winner in 1997. He also wins the climber's polka dot jersey three times (1996-98).

Spain's **Miguel Indurain** (a hero of Wiggins' youth) wins five consecutive Tours from 1991 to 1995. Riders from the ex-Communist bloc start making the top 10 from 1993 on.

The first wins for English-speakers come in 1986 (**Greg LeMond** of the US) and 1987 (Ireland's **Stephen Roche**). LeMond goes on to win twice more, while Roche's compatriot **Sean Kelly** wins the green jersey four times. **Robert Millar** is the first Briton to win a jersey - the climber's polka dot jersey - in 1984.

France's **Bernard Hinault** wins five times between 1978 and 1985 - matching the French record set by **Jacques Anquetil** 20 years earlier. No French rider has won since.

The legendary **Eddy Merckx** ushers in a period of Belgian dominance by winning the Tour, and the green and polka-dot jerseys, in 1969. Five Belgians make the top 10 in 1970. Merckx wins four Tours on the trot and Belgians end

Figure 4: Colour icon visualisation for categorical data [4]

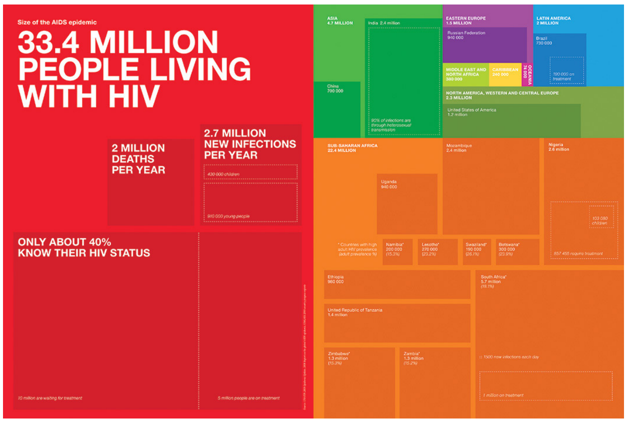


Figure 5: Treemap visualisation with three levels [5]

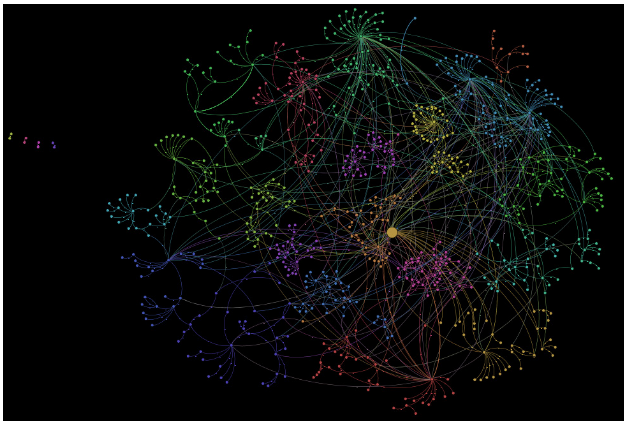


Figure 6: 3D graph visualisation with multiple branches [6]

2.4 Existing urban simulation systems

There have been attempts in the past to create a stronger connection between urban analysis tools and CAD drawing applications: among them are the systems of Cityscape, Urban 5 and Urban Simulation.

The development of Cityscape took place in 1970 at the University of California of Los Angeles (Liggett, Jepson, 1994, p. 292). The project serves as an early prototype of what later would become the Urban Simulation System. Cityscape aimed for a realistic three-dimensional simulation of communities and neighbourhoods in the city of Los Angeles, using a combination of aerial photographs and street level footage to generate the 3D model. Its original use became irrelevant with the arrival of 360 degree photography and commercial services like Google Street View.

In the same year, the Urban 5 project finished at the Massachusetts University of Technology (Batty, Dodge, Jang, Smith, 1998, p. 3). The product forms the fruit of the highly competitive Computer-Aided Urban Design class. The software served as an early attempt to use computer-aided design as a template for design instead of a supporting feature. One of the designers of the Urban 5 system, Nicholas Negropont, became a proponent of the agent simulation method in later stages of his life.

The Urban Simulation System came to life in 1995 by the hand of the creators of the Cityscape project and served as a successor of that system (Liggett, Jepson, 1994, p. 294). The project aimed to link Geographic Information Systems, mundane tools in the field of social geography sciences, with CAD tools, instruments well known by architects and engineers. The integrated tools within the Urban Simulation System were an updated version of the Cityscape project, Multi-Gen, AutoCAD (release 12) and ArcView. The Cityscape component interpreted photo- and video footage from areal- and street level to building volumes. The three-dimensional modelling environment Multi-Gen allowed incorporation of photo-realistic imagery. AutoCAD sets the structure for CAD and drafting interaction, due its frequent use in the field architectural and urban planning. ArcView offered a platform GIS analysis. The foundation of the model were processed in CAD format, later combined with photo imagery by the new Cityscape engine. These two sources create a two possibilities for further analysis: one for raw data analysis in GIS software and one for the real-time 3D visualization of the geometry.

The end of the 20th century showed significantly less progress in the development of an urban information system directly aimed at the architectural design process. In the last decades, GIS techniques accelerated in their development and importance for the social- and physical geography sciences. Still, GIS seems unfit for tools that need accuracy at an architectural level, like traditional CAD tools and BIM software.

One of the more recent researches aimed at the integration of GIS technology with generative architectural design methodology originates from the Technical University of Delft (Beirao, Duarte, Stouffs, 2007, p. 930). The research does not offer a working prototype, though grants a theoretical explanation about the ideal structure of such design work-flow. The top right figure on the next spread portrays that data flow model.

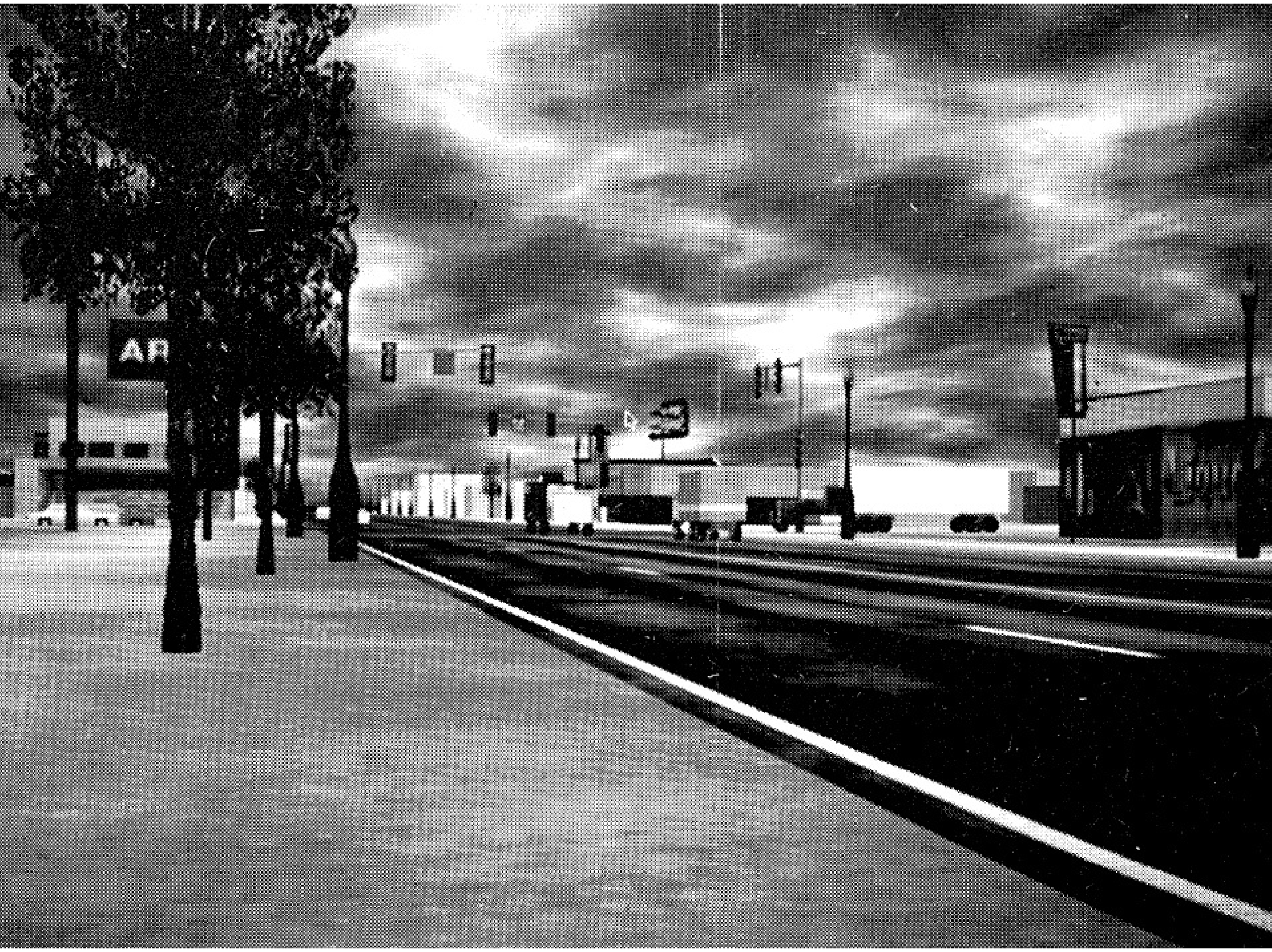


Figure 7: A Cityscape model of South Los-Angeles [7]

This model contains four stages. In the first stage, the user collects pre-existing data (in this case presented as GIS files) and inserts this data in the software. The second stage, meanwhile the first stage that the software executes, carries the name of the programmatic modelling phase. The programmatic model interprets the pre-existing data and converts them to readable patterns. The first part of this stage develops the formulation model, which prepares and converts the data to readable information for the software. The second part of the programmatic model applies this information in the urban program model. This part executes the analysis and points out the urban patterns. The software inserts these patterns in a generation model, that serves as the third phase of the model. The generation model generates new urban tissue using the patterns provided by the programmatic model. These actions result in designs that need to be manually checked and adapted by the architect.

This model also makes an important distinction between data attributes and classification. Data attributes assign data values to the objects, such as geometric volume, number of floors, building age, function, among others. Classification alter the behaviour of the object during the architectural generation process. These classifications marks human interpretations as absolute descriptions about the object. For example, an object can hold the label of historical value or a landmark. These classifications play an important role in the generation model, as they can serve as obstacles or attraction points and influence the outcome of the generation significantly.

The same research from the Technical University of Delft also identifies a fixed set of object classes. Objects that are formed in the formulation model phase will be categorised in one of these classes. These are displayed in the figure on the bottom right.

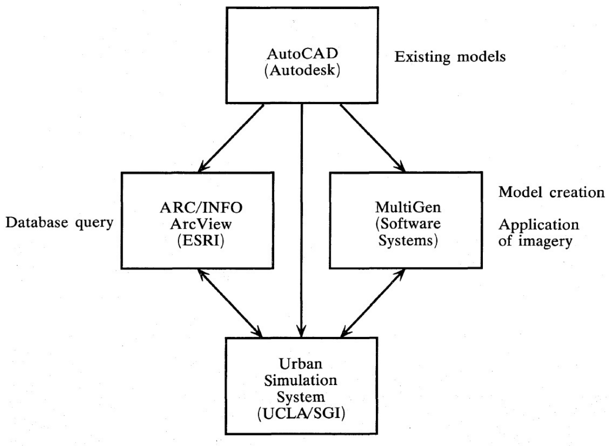


Figure 8: Software that the Urban Simulation System unites [8]

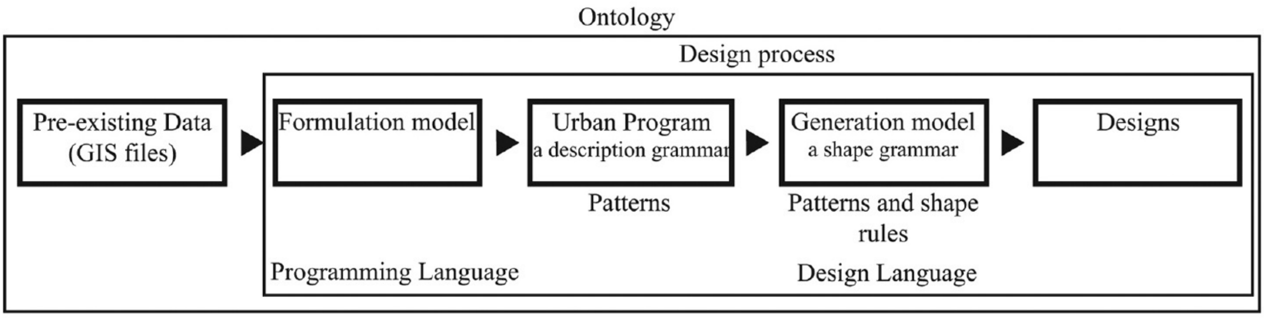


Figure 9: The data flow model[9]

Abbreviation	Object class	Description
TO	Topography	Physical landscape surface attributes
SM	Site morphology	Geometrical landscape attributes
NF	Natural features	Non-artificial elements, like vegetation or elements constructed by fauna
Z	Zoning	Administrative regions
TN	Transportation network	Roads and other traffic-related infrastructure
SQm	Main squares	Squares with a high quantity of visitors
UP	Urban plots	Urban real estate property that is available for development
SN	Street networks	Urban space that is used as logistic infrastructure
UU	Urban units	Community area, as defined by the INSEE
BL	Block	Building blocks
SQ	Squares	Squares with a low quantity of visitors
BD	Building definitions	Building properties
UF	Urban furniture	Furniture objects in public space
MA	Material	Only applies to architectural, infrastructure and urban objects
FC	Façade constraints	Describe the building’s exterior properties

Figure 10: Object classes as defined in the data flow model

2.5 Urban 3D data

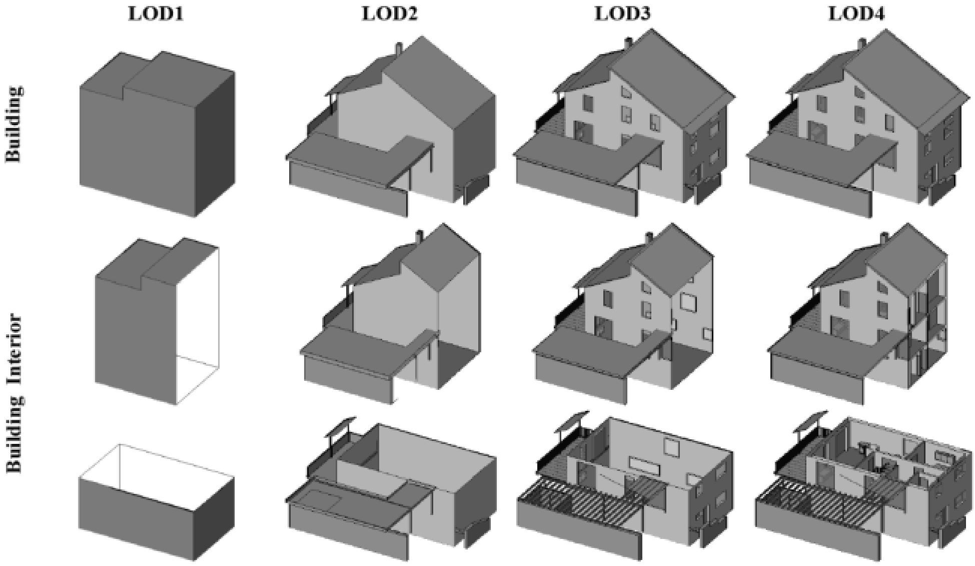
Most of these urban simulation tools rely on 3D data for their visualization. 3D data can contain different Level Of Detail (LOD) values, which range from 0 to 4. Examples of these LOD levels are displayed in the figure on the right.

The designer can retrieve 3D data in various ways. First, he can use external databases for 3D data extraction. The designer can also choose to extrude 2D CAD data to building volumes by hand in the case of 3D data scarcity, creating a LOD 1 3D model in the process. Several other 3D model generation methods do exist to generate 3D models with a detail level of LOD 2 or higher. Three well-known work-flows are LIDAR data interpretation, 3D model generation using video reconstruction and aerial footage interpretation.

Light Detection and Ranging (LIDAR) data serves as a primary data source for the generation of a Digital Terrain Model (DTM) and 3D city models (Zhou, Song, 2004, page 347). LIDAR data contains point cloud data from laser scans, offering a valuable source for an accurate resemblance of the real world. The data often comes with large file sizes, making it necessary to apply interpolation algorithms for cloud point reduction and file-size compression. The scans only offer geometric information about the analysed surfaces and make no classifications. Buildings, landscape and flora consist of the same 3D matter and need separation by the designer or interpretation algorithms. Other sources, like aerial imagery or 2D CAD data, can influence this outcome in a positive manner. Street captured LIDAR data can offer untextured 3D models up to LOD 3. The scans can also be performed from interior spaces. This makes LIDAR data a valuable for 3D model generation, independent of the required LOD level. CAD data can help with the classification of the object defined in the LIDAR data

Video footage can also serve as a base source of information for urban 3D model generation (Pollefeys, Nister, 2008, p. 144). In contrast to LIDAR processing, the video footage needs to be complemented with matching GPS data to generate an accurate 3D model. Both LIDAR and video interpretation methods can utilize ground- and drone footage. Drone footage requires a third “GPS” coordinate in the Z-direction in the form of an altitude value once used for video interpretation. 3D model generation using video footage makes use of the colour value of the individual pixels in the video image for its interpretations, making light intensity and light changes an additional factor for the quality of the outcome. This process result in a point cloud that work with the same interpolation processing methods for LIDAR data. Video data can also generate building textures and complement untextured 3D models from LIDAR data.

Aerial footage interpretation utilizes satellite or drone imagery. Several application strategies for 3D data extraction exist: DSM data matching, images comparison for the same subject and shadow interpretation. All these strategies share the disadvantage that the extraction process becomes more complex when the density of objects increases. 3D data extraction for dense urban areas ask for more source material than aerial footage.



Level	Scale	Description
LOD 0	Regional Model	2D surface without height
LOD 1	City/Site Model	Block model without roof shapes
LOD 2	City/Site Model	A textured LOD 1 model with differentiated roof structures
LOD 3	City/Site Model	A LOD 2 model containing 3D architectural elements (windows, doors, etc.)
LOD 4	Interior Model	Fully “walk-able” architectural models

Figure 11: A visualisation of LOD steps for urban 3D models [10]

2.6 GIS and its disadvantages

GIS disqualifies as a practical platform for automated architectural design. Though these tools have developed rapidly in the past years, GIS software still copes with disadvantages regarding data availability and software complexity (Batty, Dodge, Jang, Smith, 1998, p. 33).

Unfortunately, only a limited amount of GIS data is available to the public. This applies the most to projects in rural areas. A part of the available GIS data requires with a paid license due commercial interest. This means that the design team that wants to use this data needs a high budget for the data licenses.

The GIS data might contain inaccuracies, even when the retrieval of the data succeeded without a flaw. GIS has been developed as a tool for geospatial analysis and cartography, these tools neglect a real distinction between geographic and geometric information. High resolution data, with an accuracy of 50 meters or preciser, is rare. The data presented contains clusters based at postal codes or area district, a detail level to fails to meet the accuracy requirements for analysis in at architectural scale. Most GIS software also lack 3D capabilities, which makes GIS tools unfit for the 3D-reliant field of architecture.

GIS also requires expertise on database management and cartography to reach its potential. It is questionable if it can be asked of architects to learn these skills on top of the already broad variation of skill-sets that an architect needs to master to perform his duties properly.

The recent tool CityGML tries to solve these problems (Buuksahlih I, Isikdag U, Zltanova, 2013, p. 22). This XML based GIS system has native support for 3D geometry. It offers compatibility with familiar 3D software for architects, like Autodesk 3Ds MAX and McReel Rhinoceros. Though, this system does not cope with the remaining data availability and management complexity problems.

The architecture field would benefit from an accessible analysis tool that relies on accurate, easy obtainable urban data.

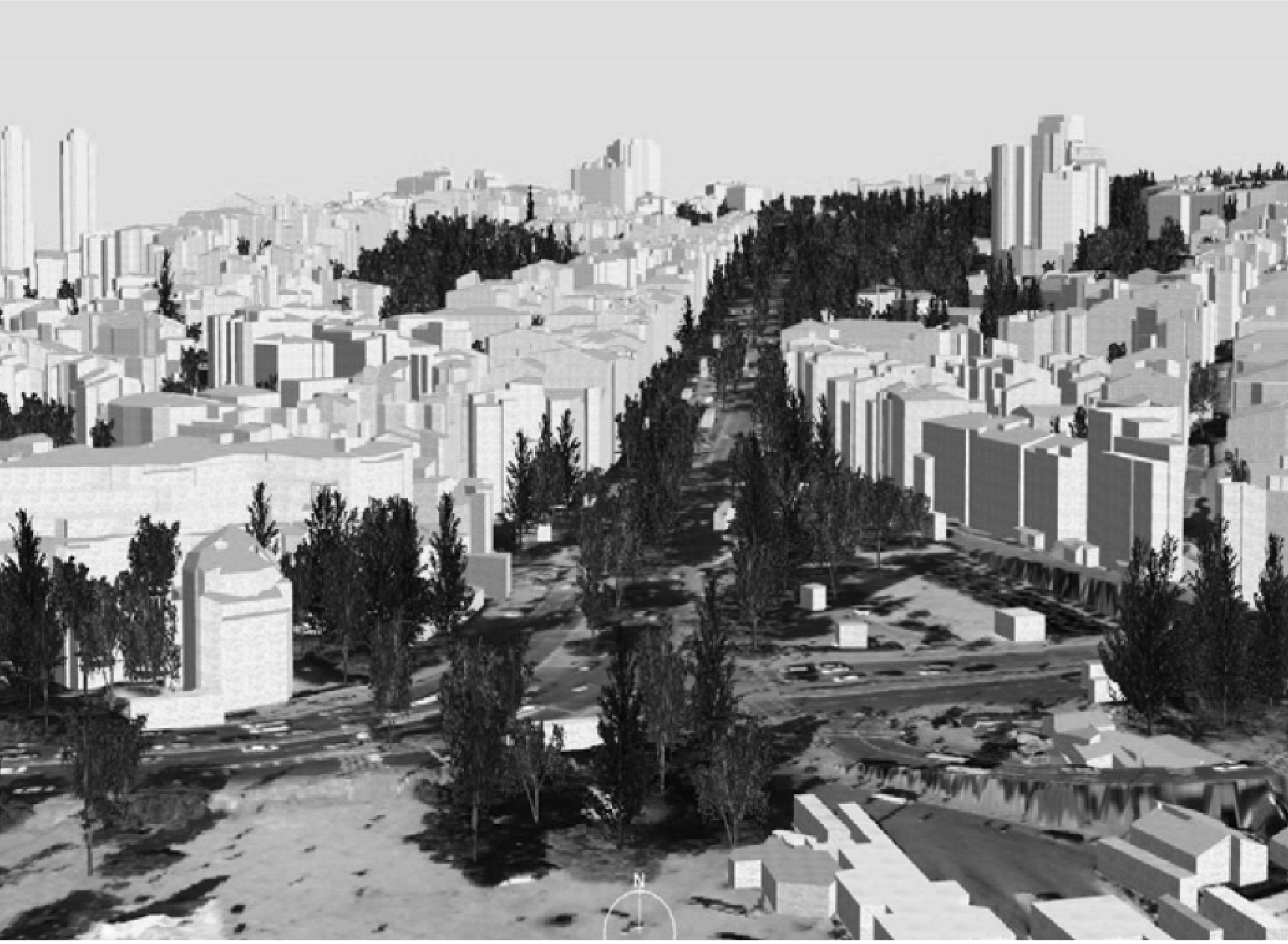


Figure 12: A CityGML model of Istanbul at LOD 2 detail level [11]

3.
REQUIREMENTS

3.1 Work-flow

As described in the relevant work section of this thesis, the data flow model (Beirao, Duarte, Stouffs, 2007, p. 930) divides the ideal set-up of the computer-assisted design process in a data collection, pattern analysis and design generation phase. This data flow model will serve as a guideline for the development of the work-flow of the urban analysis tool. The architect has responsibility over the first phase of the data flow model: the collection of pre-existing data. The urban analysis tool only aims to offer urban insight and lacks the ambition to generate design propositions. Therefore, the tool will only cover the pattern analysis phase of the data flow model.

At first, the architect needs to obtain basic information and conditions about the project. The location of the project site, expected building program and approximate budget are good examples of useful basic information. The urban information tool needs 3D data of the surrounding environment of the project site and GPS data that will link the 3D data to the real-world scenario. The acquisition of the 3D data succeeds with the manual build, automatic generation or external database retrieval of the 3D model. This means that the urban analysis tool only uses a part of the information that the architect requires to develop the conceptual architectural design. The building program does not play a role in this urban analysis phase.

An optimal work-flow of urban analysis tool contains a minimal amount of manual interactions by the architect. The final results of the analysis appear once the analysis computing has finished, the export procedure prepares interesting results for use outside the urban analysis tool and application in later design process phases. The figure on the right marks the duties of the urban analysis tool as “analyse urban context” and divides them in three steps. The software starts with integrating the 3D data with additional information about the project site’s surroundings. This step, named the data analysis, fills data gaps in the pre-existing data with information from external databases. The urban analysis tool may fill the data model with information about the function of services in the building, location public transportation stations and time of activity in the buildings. The tool will proceed with the step of data visualisation once it finalized the results of the data analysis. Data visualisation isolates data groups to assist the architect in finding helpful urban patterns for the development of the problem definition or conceptual architectural design. The architect can export this particular data to external software.

At this point, the urban analysis tool has fulfilled its duty. The architect can use the new obtained data directly for the development of a problem definition or the conceptual design. The architect could

work-flow of the design process with the urban analysis tool

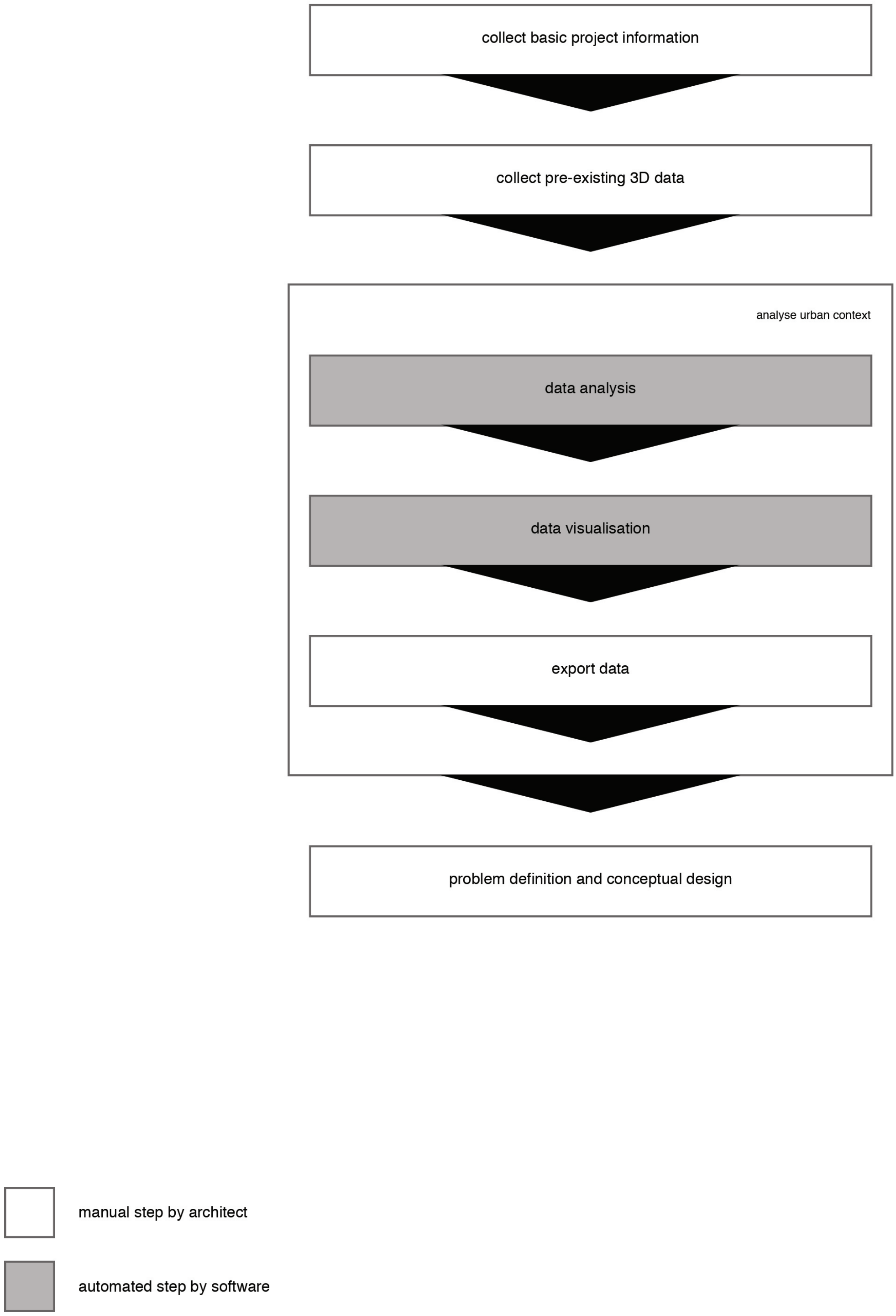

collect basic project information
collect pre-existing 3D data
analyse urban context
data analysis
data visualisation
export data
problem definition and conceptual design
manual step by architect
automated step by software

Figure 13: Work-flow diagram

also choose to perform analysis on other sites to compare the respective architectural qualities of the urban tissues. Analysis for other sites starts over the process, as it bases around other pre-existing data, and requires to walk through the work-flow diagram from step one. The design of the urban analysis tool focusses at Development Sustainability Processing, having Site Selection as a secondary objective.

3.2 3D data retrieval

The urban analysis tool benefits from the use of 3D data that the architect has already collected. In a traditional architectural design work-flow, the architect gains insight in the spacial qualities of the urban tissue with the use of a virtual or hand-crafted model. The urban analysis tool profits from these activities when its data structure bases around a digital 3D model. This data model can grow with the addition of external data. A 3D model of LOD 2 level of higher is preferred. The architect might be unable to acquire a 3D model at this detail level. In this case, he can replace the 3D data with extruded 2D CAD data. Height maps, reference picture or location visits offer clear information about the required extrusion heights. Extruded 2D CAD data result in a 3D model at LOD 1 detail level that the urban analysis tool can use as an alternative to LOD 2 data.

The support of 3D file formats from commonly used 3D software packages will ease the architect in the 3D data preparation, as the analysis tool will not directly provide with 3D data. The architect needs to export the 3D data from his common used 3D software to the urban analysis tool.

3.3 Data analysis

The analysis tool should complement the 3D input data with the information that has a direct influence at the architectural design. The urban analysis tool might prove useful in the scenario of the architectural competition, so it should execute the analysis that will aid this purpose. Topics that may conform as interesting topics of analysis are geometric qualities of the building volumes and landscape elements, view axis, functions of services located in the urban area, time and intensity of activity in the area surrounding the project site and public transportation layout. This thesis will focus at analysis of the cultural aspects rather than the analysis of the geometric qualities of the urban surroundings, as no tool exists yet that can full-fill these tasks with the amount precision that architectural design requires. This leads to the conclusion that the urban analysis tool should manage to gather information about function of services in the area, amount of services per building block, opening hours for these services and the location of public transportation stations near the project site.

3.4 Data visualization

The visualization of the data should resemble architectural drawings, as the architect is familiar with reading them. The urban analysis tool should present its visual output to the architect in a clear manner. The goal of the visualisation is exploration: finding patterns in the data to develop a hypothesis about the data. This means that the data visualisation should have a high level interactivity. The user should have the option to filter out unnecessary data to isolate the required data for helpful conclusions about the surroundings of the building site.

3.5 Exporting data

The urban analysis tool should offer the possibility to export its internal data for use in external software packages to increase its usability in the architectural design work-flow. 2D data, 3D data and sheet data could assist the architect in the further design work-flow.

4.
SYSTEM DESIGN

4.1 System architecture

The user will interact with the urban analysis tool through a user interface. This user interfaces controls the assets in the software that require input of the user, such as the set-up of the visualization filters. It also offers visual feedback to the user about the data stored in the tool and the status of the import- and analysis functions. These functions will result in the generation of an Urban Information Model, an object that contains all the information about the urban environment of subject. The visual output of the user interface displays this new information. The Urban Information Model consists of several subgroups of informations, each representing different type of data elements in the urban environment. The categories follow the guidelines as presented in the data flow model as presented in the related work section of this thesis.

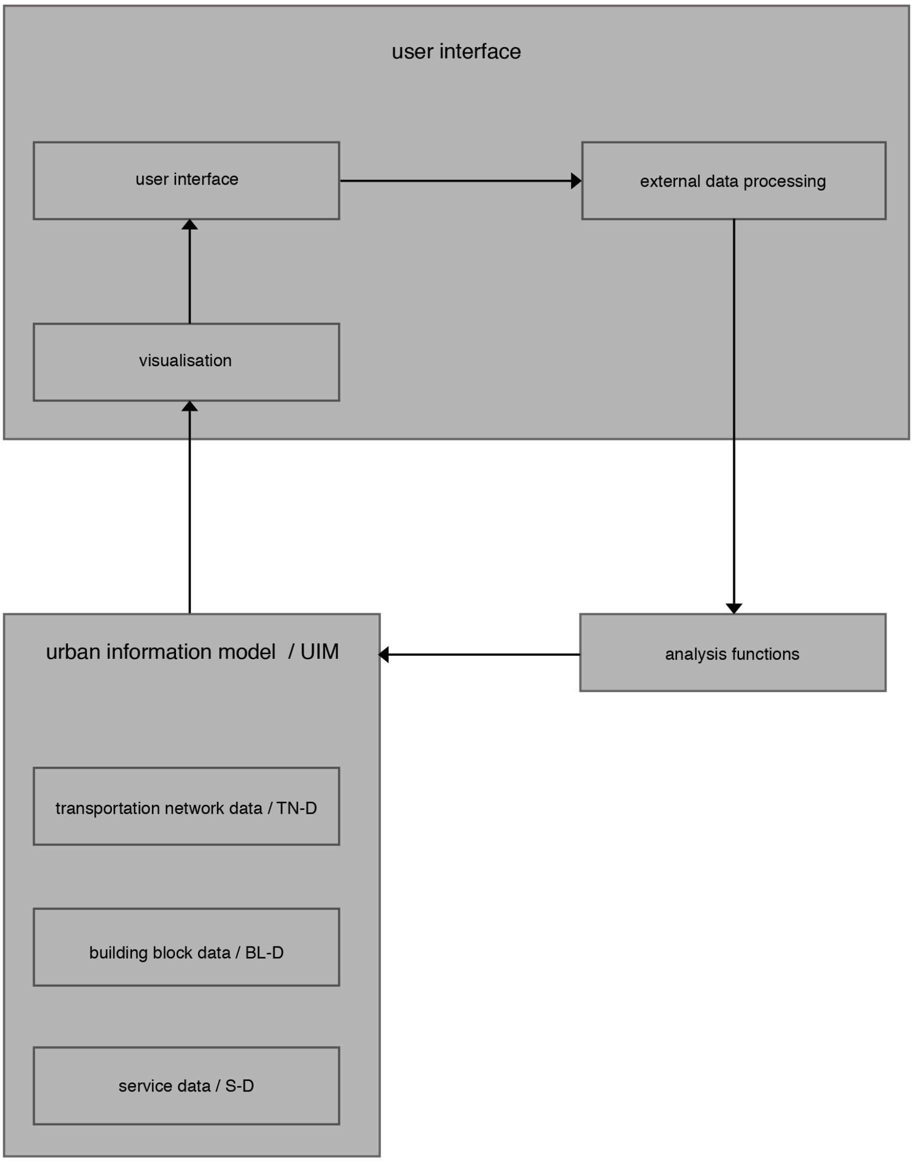


Figure 14: Conceptual system architecture

4.2 Conceptual work-flow

The architect should check the 3D data for errors and convert it to a file type that the urban analysis tool can understand, before the software can import this pre-existing data. Possible errors in the 3D data can come in the form of broken meshes, unnecessary (guide) objects, origin point offsets, incorrect scaling and incorrect orientation. The urban analysis tool also requires a connection between a global and local coordinate to link the virtual 3D model to real-world information. A GPS coordinate defines this global coordinate. Step 3 and 4 in the figure on the right represent these phases of data preparation.

Subsequently, the urban analysis tool imports the checked pre-existing data upon start-up. The tool creates a basic urban information model with this information, using the 3D data that has been collected as pre-existing data before. The software first tries to complement as much data as possible with the information that model already holds. In praxis this means that the urban analysis tool links address data to the individual meshes. The figure on the right marks this phase as “static analysis”. When the model still contains data gaps after this step, it will will receive complementing data from external (online) databases. This step, marked as “dynamic analysis”, occurs after the completion of the static analysis. The urban analysis tool combines the data from the dynamic analysis and static analysis for the generation of new data once both analysis phases have finished. This new information offers the architect new, possible useful, urban patterns to the architect. The software sequence ends with the steps of data visualisation and data export, as described in the work-flow figure in the requirements chapter: Data visualization isolates information that prove useful to the architect, data export extracts this information for later use in external software.

The architect can use the new information in other software packages to develop a problem definition or conceptual design.

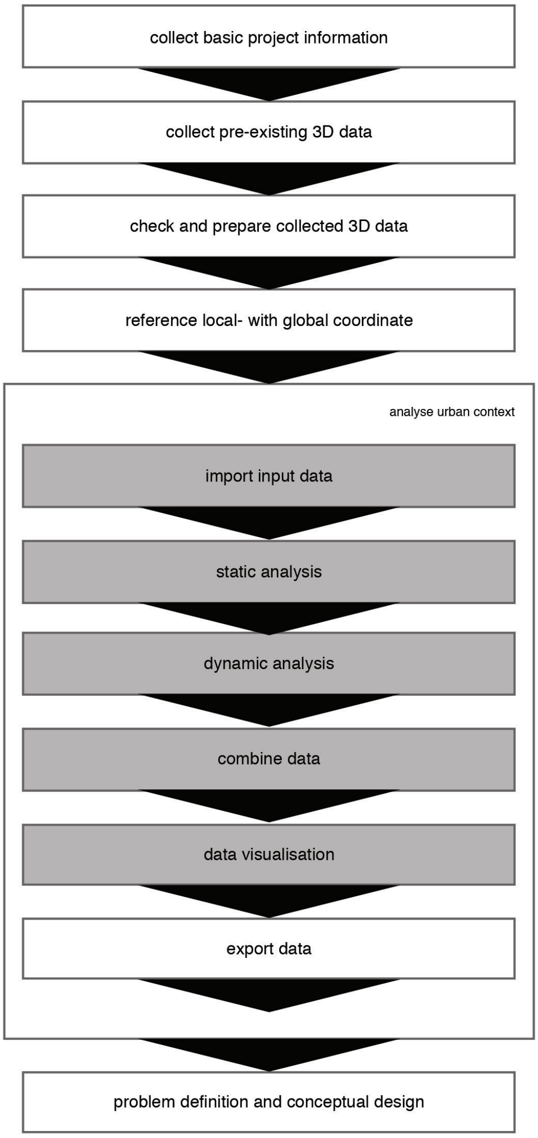
4.3 Input data

As the urban analysis tool will not directly provide the 3D data, therefore it should support 3D file formats that allow editing by architects in external 3D software. Examples of commonly used 3D software packages in the architecture industry are 3Ds MAX, Blender, Rhinoceros, AutoCAD and Revit. The 3DS, OBJ, DXF and STL file format offer good options for 3D file formats that work well with these software packages.

The requirement of compatibility with the import modules of the urban analysis tool draws a second condition for the chosen 3D file format. This means that the chosen 3D format should have compatibility with the external 3D software packages and the urban analysis tool at the same time. Importing textures remains trivial and unnecessary, as the urban analysis tool merely requires the 3D data to learn about the geometry of the urban environment.

The architect should check the 3D data manually before he can use it as pre-existing data for the urban analysis tool. The 3D model can contain deviation from the real world scenario, especially if the architect has build the model from scratch. A retrieved 3D model from an external source may hold unnecessary guide objects or broken meshes.

The urban analysis tool requires more than 3D data to generate a “smart model”. It needs a connection to the real world scenario, as the 3D data only contains information about its virtual coordinate system. A single link between a virtual and global coordinate solves this connection problem. A geolocated address serves as a good alternative to the global coordinate, in case one is unavailable.


collect basic project information
collect pre-existing 3D data
check and prepare collected 3D data
reference local- with global coordinate
analyse urban context
import input data
static analysis
dynamic analysis
combine data
data visualisation
export data
problem definition and conceptual design

□ manual step by architect

▨ automated step by software

Figure 15: Conceptual work-flow diagram

4.4 Static analysis functions

Static analysis functions only require information that does not need frequent update to stay relevant. The architect has the option to retrieve this necessary data by hand, as the results of these analysis have a low risk of deviation from the real world scenario due experation. Nevertheless, retrieving the data this way might lead to a substantial amount of manual labour. Local-to-global coordinate conversion, floor-area ratio calculation, architectural grain size analysis and isovist curve generation illustrate examples of static analysis functions. The urban analysis tool as presented in this thesis will only use the static analysis function of local-to-global coordinate conversion. A significant amount of research about the application of static analysis functions within urban context is already available, making the application of dynamic analysis functions a more interesting focus point for the first prototype of the urban analysis tool. However, the system architecture of the urban analysis tool leaves space for the other static functions to be implemented in future versions of the software.

4.5 Dynamic analysis functions

Dynamic analysis functions depend at information that needs to be updated frequently to stay relevant. The architect can initiate this update process manually, though direct data retrieval from external (online) sources result in less manual labour and less data transfer errors due human acting. The dynamic data retrieval process demands the use of an application programming interface (API) to interact with the external database. Most public available data sources come with their own API.

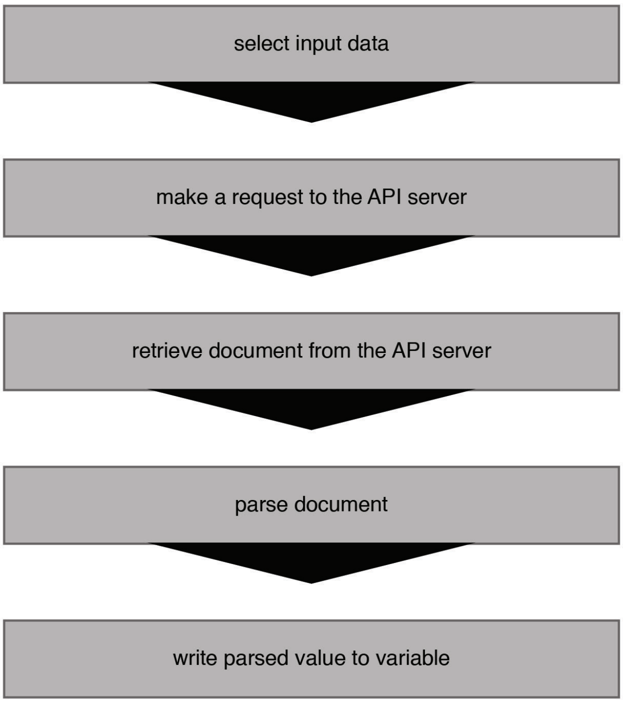
The standard data retrieval procedure for most APIs succeeds with five steps. At first, the user or the software needs to assign to assign pre-existing data as input data for the request. The software will make a request to the API server with this given input information. The API server returns a document. The contents and file format of this document depends on the API and the type of input request. The API may return more information than the urban analysis tool requires. The software has to parse the returned document to search for the relevant information. Then, the tool writes this information in a variable in the code or saves the data on the local disk or data cloud.

The two dynamic analysis functions that the urban analysis tool uses are “geocoding” and “service function retrieval”. Geocoding converts a known address to a valid GPS coordinate. Reverse Geocoding opposes Geocoding and converts a GPS coordinate into an address. Service type retrieval obtains information about services that are situated in the building , defining what public role that building plays. The exact structure of these functions is explained in the implementation section of this thesis.

4.6 Combining data

Combining the results of static and dynamic analysis functions could give the architect new insights in the architectural quality of the given urban environment. Data interpretation that might prove useful for the design process are service function mapping, time of activity and service quantity in the building volume.

Service function mapping measures the distance from the respective building volumes to the nearest service with the function of choice. Examples of service functions are education services, commercial areas and health institutes. The distance between two objects can be calculated as the crow fly distance or the result of a path finding algorithm. The calculation of the time of activity succeeds with the comparison between the building service’s opening hours data with the model’s global time and date. The urban analysis tool generates information about service quantity in the building by counting the amount of services that that volume holds.


select input data
make a request to the API server
retrieve document from the API server
parse document
write parsed value to variable

□ manual step by architect

▨ automated step by software

Figure 16: Standard procedure for executing API requests

The urban analysis tool calculates general information about the model, like building volume count, amount of services in the area, amount of missing data and averages of available data, and writes these in separate variables and sheets to aid the user in the comparison of different urban tissues.

4.7 Data visualization

The tool will make use of a wide and bright palette consisting of easily distinguishable colours to make the images clear to read. The visual feedback will also contain information about the applied data filters and a legend to ease architect in the output interpretation process.

4.8 Exporting data

Data export represent an important asset of the analysis tool, as it connects the urban analysis tool with external design software. The urban analysis tool will have the opportunity to export 2D data, 3D data and sheet data. 2D data export can result in raster- or vector data. Common file formats for 2D raster data are JPG, PNG, BMP, TGA and TIFF, common file formats for 2D vector data are DWG, SVG and EPS.

The urban analysis tool will not add new 3D data to the model. Therefore, potential exported 3D data needs to contain the addition information to be different than the imported 3D data, with the dynamic analysis output as the most useful data for the conceptual design development phase. An addition of colour coded textures can form a valid alternative to symbolise this information within the exported 3D document. 3DS, OBJ, DXF and STL satisfy as valid file format options for 3D data export. The architect can extract non-geometric information from the urban analysis tool with sheet data export, using the XLS, CSV, XML or JSON file format. This thesis visualises this sheet data as graphs to improve readability.

5.
IMPLEMENTATION

5.1.1 Technical system architecture

The technical system architecture elaborates the conceptual system architecture. The user interface contains components for controlling the visualization filters and the import- and export functions that lead to the generation of the Urban Information Model object(s). The render engine reads the stored information from the UIM(s) and updates the visual input in the user interface automatically. The Urban Information model contains the same components as stated in the conceptual system architecture: Transportation Network Data, Building Block Data and Service Data. The UIM can draw from a collection of static and dynamic functions to complement its own data. The dynamic functions depend on a Google API module to grant the analysis tool the opportunity to interact with the external databases of Google. These dynamic functions initiate once the generation of the UIM object(s) has completed, as these functions complement the data that the UIM object(s) holds.

The Urban Information Model (UIM) contains all the geometry and cultural information about the part of the urban environment that the pre-existing 3D data describes. In the first version of the analysis tool, the object divides its information over three lists: Building Block Data, Transportation Data and Service Data. This means that the UIM save information about the individual objects in these subclasses. In addition, the UIM saves information about

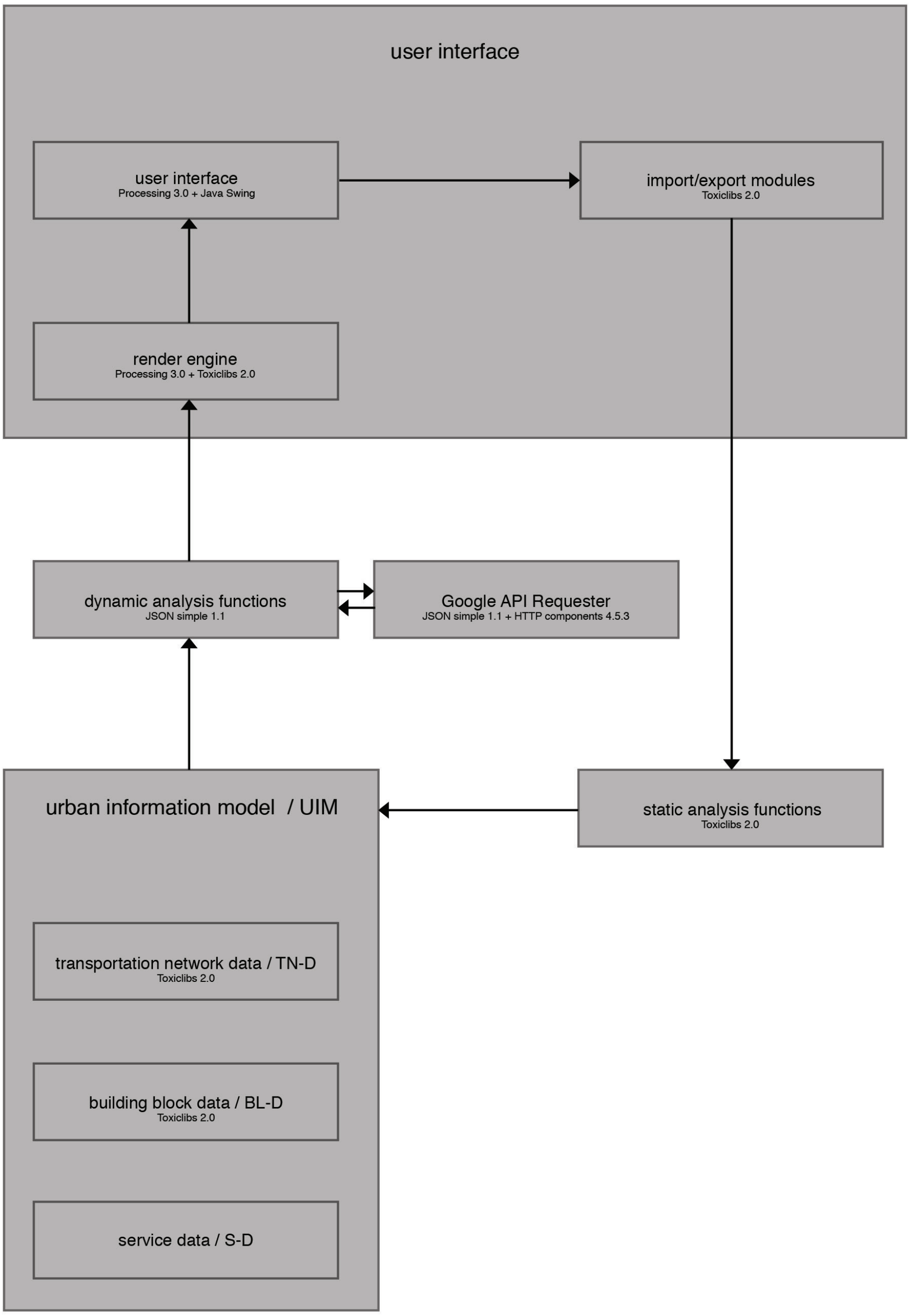

user interface
user interface
Processing 3.0 + Java Swing
import/export modules
Toxiclibs 2.0
render engine
Processing 3.0 + Toxiclibs 2.0
dynamic analysis functions
JSON simple 1.1
Google API Requester
JSON simple 1.1 + HTTP components 4.5.3
urban information model / UIM
static analysis functions
Toxiclibs 2.0
transportation network data / TN-D
Toxiclibs 2.0
building block data / BL-D
Toxiclibs 2.0
service data / S-D

Figure 17: Technical system architecture

the global urban tissue as its global variables. An example of these global variables is the global-local coordinate link.

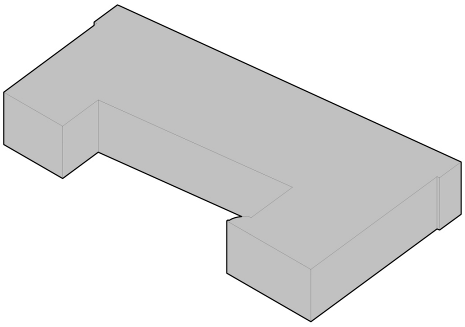
The UIM object attempts to fill in its missing information upon initialisation, using a combination of static and dynamic analysis functions. This means that the UIM not only serves a data container, but also initiates the necessary calculations and API requests for its data completion.

The urban analysis tool only contains one UIM when the tool in the process of Development Sustainability Processing, but also offers the possibility to load multiple UIM objects to support the process of Site Selection.

Building Block Data (BL-D) contains all the information that is related to the building volumes in the UIM object. BL-D only serves as data container and therefore lacks the capability to perform analysis functions on its own. The software triggers these functions from the UIM the BL-D object belongs to. A BL-D object contains a mesh, an address and a Service Data (S-D) list for that specific building.

Unfortunately, most 3D information at LOD 2 level or lower only define individual meshes for building objects. This means that vegetation and public transportation objects have no representing mesh in the 3D model. Using a mesh to store information about the position of public transportation objects will therefore lead to a deviation from the real world scenario. A division between Building Block Data and Transportation Network Data (TN-D) solves this accuracy problem. The locations of a TN-D object depends on a local coordinate instead of a mesh object. A TN-D contains a local coordinate, a global coordinate and information about the type of transportation node.

Service Data (S-D) objects merely function as data containers, holding information about a public service: its service type, address and time of service activity. Due the system architecture of the chosen API, the UIM creates all the S-D objects upon initialisation its, then creates instances of the S-D objects for their corresponding BL-D objects using address matching. This means that the UIM and its BL-D objects contain a list of S-D objects. The urban analysis tool labels the list of S-D objects that relate to a BL-D object as local S-D objects, the list of unassigned objects S-D objects that therefore relate to to the parent UIM object as global S-D objects.



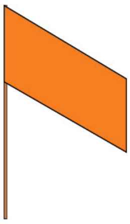


Figure 18: A representation of a BL-D object

Figure 19: A representation of a TN-D object

5.1.2 Run-time environment

The analysis tool will be developed as a stand-alone application with Java as its main run-time environment. The Java coding language bases on objects building and therefore holds the possibility to expand the urban analysis tool once the first prototypes has proved to work. Java also offers a wide range of libraries that can handle important actions for the urban analysis tool, from 3D model import to HTTP requests, and offers the possibility to import previously written object classes and libraries into the main project with ease. A stand-alone application makes it easy to incorporate the software in a standardised design work-flow, as it proves less dependent at pre-installed software and could be written to prepare the export data for a good correspondence with a broad package of external design software.

5.1.3 External libraries used

The Java language has a wide range of build-in functions, but needs to be expanded with libraries to execute all the import- and analysis functions of the urban analysis tool. Therefore, the urban analysis tool will complement the standard Java JRE libraries with a collection of graphic, vector and data request libraries.

The external libraries that have been used with the first version of the urban analysis tool are displayed in the figures on the right.

Java library	**Description**
Processing 3.0	A multifunctional library aims to make the Java language more accessible to visual artists. The library offers a strong frame for calculating and displaying geometrical objects and text, 2D and 3D (OpenGL) visualizations and sheet file in- and export
Toxiclibs 2.0	A vector calculation library that elaborates on the Processing library. The library also offers strong 3D mesh editing tools and .STL import modules
HTTP Components 4.5.3	A library for initiating HTTP requests to external servers
JSON simple 1.1	A for handling and parsing JSON documents

Figure 20: Used Java library descriptions

	Processing	Toxiclibs	HTTP Components	JSON simple
User interface	■			
Render Engine	■	■		
Import/Export	■	■		
UIM				
TN-D		■		
BL-D		■		
S-D				
Static Analysis		■		
Dynamic Analysis				■
Google API			■	■

Figure 21: Application overview of Java libraries in the urban analysis tool

5.2 Technical work-flow

The technical work-flow elaborates of the conceptual work-flow. The 3D data of surrounding environment of the site requires a detail level of LOD 1 or LOD 2 to work well with the urban analysis tool. The user needs to save the 3D data saved in STL file format, as this document type that conforms well with the implemented 3D libraries of the urban analysis tool. The user needs to save an additional TXT document, containing the link between the local and global coordinate needs. He can elaborate the text document with ID numbers of the 3D mesh objects that will symbolise the project site in the urban analysis tool.

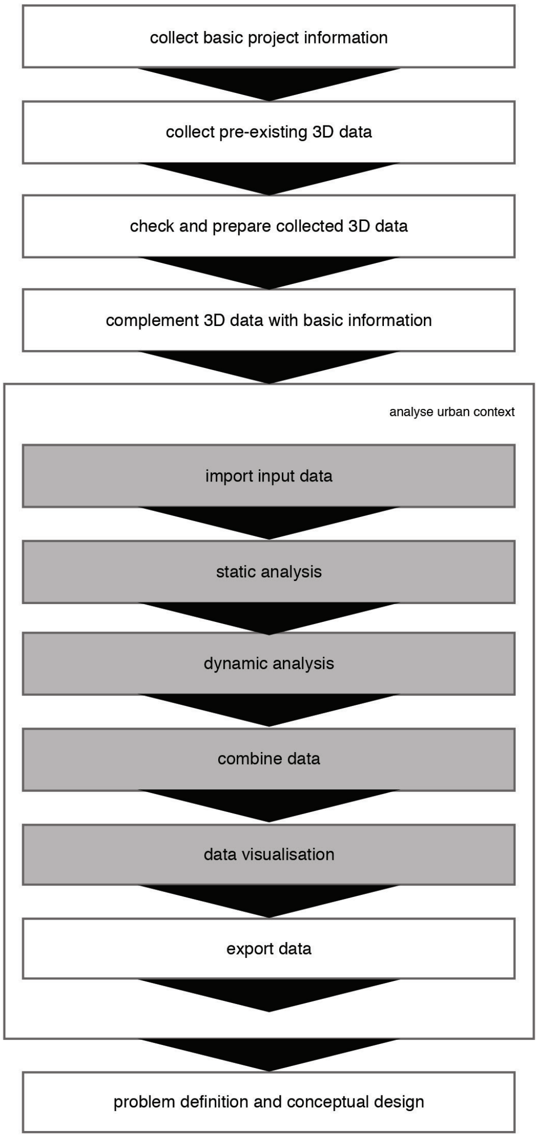
The imported data act as a foundation for the generation of the Urban Information Model (UIM). This UIM object contains a global-local coordinate link and a Building Block Data (BL-D) object for each imported STL mesh. It will also marks the BL-D object(s) that the TXT document defines as project site. At this point, the BL-D objects only contain a mesh object. Additional information within the BL-D still lacks.

The static analysis complements the BL-D objects with address data using Williams Aviation formula and reverse geocoding (both explained in §5.4.2). It also merges BL-D objects that share the same address into one BL-D object. The dynamic analysis retrieves data from the Google servers using the Google Maps API. These actions generate a global service data (S-D) list for the UIM object. Duplicate S-D objects will be removed. The global S-D objects append to local S-D lists of their belonging BL-D objects using address matching. The combined outcome of the static and dynamic analysis functions result in the generation of new information that may assist the user in the search for useful urban patterns. The urban analysis tool can export the collected information within the UIM as as raster- vector- or sheet data.

5.3 3D preparation, import and limitations

The first version of the analysis tool will only import the 3D data and therefore not generate it on its own. This means that the user needs to prepare the retrieved 3D data manually before the urban analysis tool can import the data successfully. The urban analysis tool only imports STL file formats due limitations of the Toxiclibs 3D import library. The user can perform the preparation with every 3D suite, as long as that software can write the data to STL format. The 3D data for the case studies in this thesis have been prepared with Rhinoceros 5, as this software contains strong python support for 3D preparation and export automation. Due to its system architecture and technical performance limitations, the urban analysis tool only supports detail levels up to LOD 2. A LOD 0 or LOD 1 model will not influence the outcome of further automated analysis in this version of the analysis tool, as they only require 2D coordinate input. Nevertheless, the user benefits from 3D data with a LOD 2 detail level, as this could minimise the chance that the user observes the geometric qualities of the urban tissue different from the real-world situation.

The urban analysis tool generates an individual BL-D object for each 3D mesh. Therefore, the 3D data has to contain separated meshes for each imminent BL-D object. The user can execute this separation process manually within a 3D suite of choice, having the option to divide meshes based on building volume or divide based on address. Mesh division based on address will result in a more accurate generation of local S-D lists, mesh division based on building volume offers a secondary option for buildings that do not cope well with address based division. For example, multifunctional buildings benefit from mesh division based on building volume. The first version of the urban analysis tool lacks the capability to distinguish building meshes from non-building meshes. The user needs to remove non-building related geometry from the 3D


collect basic project information
collect pre-existing 3D data
check and prepare collected 3D data
complement 3D data with basic information
analyse urban context
import input data
static analysis
dynamic analysis
combine data
data visualisation
export data
problem definition and conceptual design

□ manual step by architect

▨ automated step by software

Figure 22: Technical work-flow

data, so the urban analysis tool will cease to generate false BL-D objects for vegetation or street furniture. BL-D objects that share the same address will merge into one BL-D object. The UIM will include a higher percentage of the real-world addresses if the 3D data contains smaller separated 3D meshes. In addition, the 3D model should have the axis origins as approximate centre point. This does not influence the outcome of data analysis directly, but prevents camera issues during the data visualisation phase. This correction succeeds with the translation of the complete 3D model within its local coordinate system. The user can perform 3D export process executed manually or with an export script. The STL files should carry the same name, except for the last character: this should represent the ID number of respective 3D mesh.

The user should also define the local-global coordinate link and the meshes that symbolise the project site in the urban analysis tool, splitting them with a vertical bar. The information is stored in a simple TXT document and saved in the same folder as the STL 3D data. An example of the content of this document is:

“0 | 0 | 52.373113 | 4.892831 | 74 | 105”

In this case, the local coordinate {0,0} will refer to the global coordinate {52.373113, 4.892831}, representing the centre of the Dam Square in Amsterdam, The Netherlands. The meshes with ID number 74 and 105 mark project site. The urban analysis tool can read single and multiple ID number input. The project site stays hidden if the TXT document lacks the ID information.

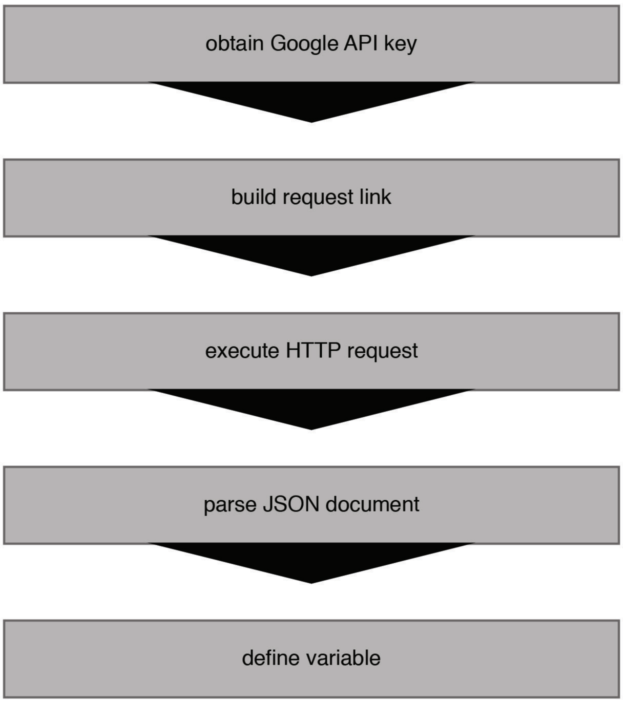
The urban analysis tool will automatically import the prepared data from its default folder upon launch. The tool will create one UIM object for the full folder, creating a BL-D object for each STL file. At this point, the BL-D object only contain the mesh data and there for lack other information. The tool also saves the local-global coordinate link in the UIM object and marks the BL-D objects that should represent the project site. The merging of address-sharing BL-D objects commence after the completion of the dynamic analysis phase.

5.4.1 Google API requests

The chosen online data source for the analysis tool is Google Maps. Google offers an elaborate API that integrates well with Java-based software like the urban analysis tool. The Google Maps gets updated frequently by Google itself, governments and business owners, making it a reliable tool for broad search of public and commercial functions in the local area. The Google Places API seems useful for the retrieval of data about these services, as a significant amount of web-based and mobile applications depend on the API for this particular reason.

The urban analysis prototype makes use of two assets of the Google API: the Google Maps API and the Google Places API. Most application developers use the Google Maps API to display interactive maps in their web- and smart-phone applications. The API supports requests for geocoding, reverse geocoding and a full database of public, commercial and transportation services as well. The API marks these services with Google Maps place IDs. These place IDs show up within the Google Maps API after the insertion of the corresponding address, though this process only succeeds if the input address matches the address from the Google Maps database perfectly. The Google Places API can find these Google Maps place IDs without this problem: it offers the possibility to locate all services of a given type in the area, reducing the chance that local services remain undiscovered due address mismatch.

Steps to complete a Google API request



□ manual step by architect

▨ automated step by software

Figure 23: Steps to retrieve information from the Google Maps API

Both Google APIs require a Google API key to work. This key can be generated at the Google Developers Dashboard and monitors the amount of requests that the software performs. The key limits the amount of these server interactions by 1000 requests per day for free licenses. Premium keys offer more interactions per day. The user could generate the key himself, but the first prototype of the urban analysis tool this key will contained a set pre-generated API keys already.

In the second step, the software needs to build a request link. The exact content of this link depends at the API and required information. All request links that interact with the Google API need to specify the desired Google API, the input information and the Google API key. The urban analysis tool generates this request link with simple string addition. An example of an API request link is:

https://maps.googleapis.com/maps/api/geocode/json?latlng=52.373113,4.892831&key=AIzaSyAWZZs6AC51GV

“geocode” specifies that the geocode module of the Google Maps API is required. “json” marks that the API should output an JSON document. The global coordinate {52.373113, 4.892831} serves as input variable. The API key in this example is “AIzaSyAWZZs6AC51GV”. Note that the API key in this example has not been registered at Google yet.

The actual HTTP request succeeds by filling an Input Steam with the response of the HTTP request. The Google APIs offer the possibility to return the data in XML and JSON format. The urban analysis tool will use the JSON file format, as the documentation of the Google API states that this alternative to be more stable and accurate.

The code snippet above displays an example of a raw JSON document that the Google API will send as result of a HTTP request. The urban analysis tool needs to filter this document before it can use its contents. Fortunately, a JSON document contains a data tree structure that eases the tool in the data retrieval process. For example, the data retrieval for the type of the service succeeds by moving through the tree over the following branches: results > types. This result in the type tags “travel agency, restaurant, food, establishment”. The urban analysis tool will store this information in the form of a string variable for later use.

The JSON example as displayed on the right only displays one result. The API response document contains multiple results, sorted by relevance. The urban analysis tool picks the first result, the most relevant result to the stated input data. As a last step, the tool will write the parsed data to one of its variables.

Large quantities of API requests will increase the start-up time of the urban analysis tool. Therefore, the amount of API requests should be kept to a minimum.

```
{
  “html_attributions” : [],
  “results” : [
    {
      “geometry” : {
        “location” : {
          “lat” : -33.870775,
          “lng” : 151.199025
        }
      },
      “icon” : “http://maps.gstatic.com/mapfiles/place_api/icons/travel_agent-71.png”,
      “id” : “21a0b251c9b8392186142c798263e289fe45b4aa”,
      “name” : “Rhythmboat Cruises”,
      “opening_hours” : {
        “open_now” : true
      },
      “photos” : [
        {
          “height” : 270,
          “html_attributions” : [],
          “photo_reference” : “CnRnAAAAF-LjFR1ZV93eawe1cU_3QNMCNmaGkowY7CnOf-kcNmPhNnPEG9W979jOuJJ1sGr75rhD5
              hqKzjD8vbMbSsRnq_Ni3ZIGfY6hKWmsOf3qHKJInkm4h55lzvLAXJVc-Rr4kI9O1tmIblblUpg2oqoq8RIQR
MQJhFsTr5s9haxQ07EQHxoUO0ICubVFGYfJiMUPor1GnIWb5i8”,
          “width” : 519
        }
      ],
      “place_id” : “ChIJyWEHuEmuEmsRm9hTkapTCrk”,
      “scope” : “GOOGLE”,
      “alt_ids” : [
        {
          “place_id” : “D9iJyWEHuEmuEmsRm9hTkapTCrk”,
          “scope” : “APP”
        }
      ],
      “reference” : “CoQBdQAAAFSiijw5-cAV68xdf2O18pKIZ0seJh03u9h9wk_lEdG-cP1dWvp_QGS4S
NCBMk_fB06YRsfMrNkINtPez22p5lRIlj5ty_HmcNwcl6GZXbD2RdXsVfLYlQwnZQcnu7ihkjZp_                    2gk1-fWXql3GQ8-
1BEGwgCxG-eaSnIJIBPuIpihEhAY1WYdxPvOWsPnb2-nGb6QGhTipN0lgaLpQTnkcMeAIEvCsS                    a0Ww”,
      “types” : [ “travel_agency”, “restaurant”, “food”, “establishment” ],
      “vicinity” : “Pyrmont Bay Wharf Darling Dr, Sydney”
    }
  ],
  “status” : “OK”
}
```

Figure 24: A HTTP response example of the Google API in JSON file format

5.4.2 Geocoding

Geocoding describes the process of converting a known address into a valid GPS coordinate. Reverse Geocoding opposes Geocoding and converts a GPS coordinate into an address.

Both vital functions have two uses: the conversion of pre-existing data to API input data and the integration of the parsed JSON information with the existing UIM model. Especially geocoding forms a useful tool for the preparation of input data for API requests. Global coordinate input surpasses address data input, as the API may return an error state if the address data contains incorrect formatting. Therefore, the urban analysis converts address data to global coordinates before using the data as input data for the Google API requests.

The vital link between local coordinates and global coordinates grants the urban analysis tool the possibility to link real-world information to the virtual entities. The software makes use of the Williams Aviation formula to calculate the relation between a local and a global translation.

This formula only calculates the relative connection to the local translation and the global translation. It requires a direct link between a local and global coordinate to return absolute coordinates. The urban analysis tool allows an geocoded address as an alternative to the global coordinate input.

Geocoding and reverse geocoding play an important role for the data complementation of the UIM object. Once the UIM is generated, it will only contain BL-D objects without any address or service function information. The urban analysis tool retrieves address information using a combination of local-to-global coordinate conversion and reverse geocoding. First, the tool will calculate the geometric centre point of the BL-D objects. These 3D points are converted to 2D local coordinates. The conversion of these local coordinates result in a coordinates that serve as input for the reverse geocoding process. The urban analysis tool writes the address strings output to their parent BL-D object.

5.4.3 Function retrieval

The analysis tool utilizes the Google Places API to find public and commercial services in the area. The API distinguishes different services functions, defining them with “type tags”. The urban analysis tool groups these type tags in type categories, as the individual type tags seem too specific to remain relevant for architectural analysis on their own. The type categories that the analysis tool distinguishes are shown on the next page.

The urban analysis tool generates the S-D objects using the Google Places API. First, it will execute HTTP requests for all type tags, using the UIM’s global centre point as input. The Google Places API will return a set of results for the corresponding type tags. These results get sorted and divided over the type tags that the UIM handles. The UIM then generates the S-D objects using this information. As a last step, the urban analysis tool links the S-D objects to the BL-D objects using address matching. Note that one BL-D object can carry multiple S-D objects, as building blocks can contain multiple public available services.

These API requests also display a unique Google ID in their output document, called the “PlaceID”. Function search based on type tag can result in duplicated responses for services with multiple type tags. Therefore, services with the same PlaceID characterize as duplicated, resulting in only the first service being saved in the UIM. The PlaceID has an alternative use for a second API request to gain more service details. The urban analysis tool needs this second request to retrieve to opening hours of the service.

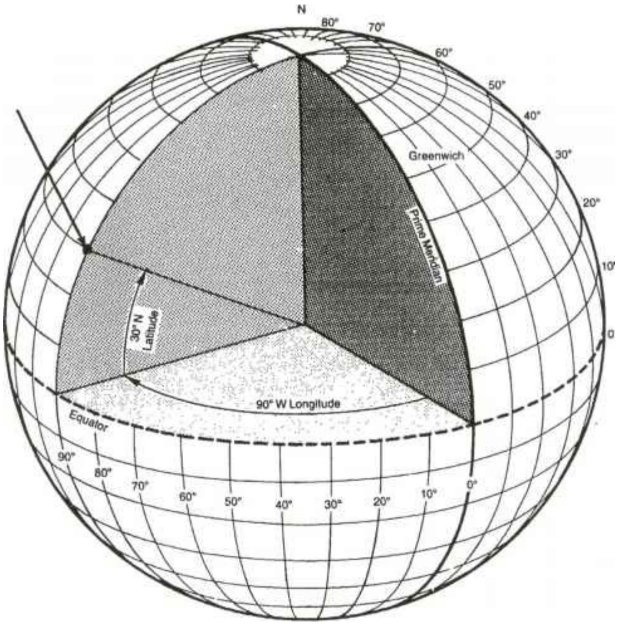


Figure 25: A visualisation of global coordinate translation [12]

The urban analysis tool can extract multiple pieces of information for the S-D object per HTTP request. Among them is information about the name of the institute, the global coordinates and the address, the opening hours, attached photos, the Google Maps place ID, the type tags (sorted to importance) and user reviews. The first version of the urban analysis tool will ignore attached photos and user reviews, but does write all other information to the S-D object as they offer convenient input data for further analysis.

The opening hour data needs further processing before it can serve as a reliable source for urban pattern finding, as the formatting of the data may differ. Examples of opening hour formatting are “8AM – 5PM”, “08.00 – 17.00” and “8 o’clock – 5 o’clock”. Different languages play a role in non-English speaking countries as well. The urban analysis tool defines the format type of the raw data, then converts this data to a standardised format for the whole data model. Opening hour data consists of an opening hour, closing hour per day of the week. This means that each S-D opening hour list contains 14 entries.

Function search using type tags has a disadvantageous dependency on the manual Google database input by the companies themselves. This means that these companies have responsibility over the description and type tag for their own service without a formal legitimacy check by Google. Type tag search may return this false information without suspicion of the user. Service that lack type tags completely remain invisible to the type tag search function, resulting in a further deviation from the real-world scenario. The urban analysis tool will performs a second round of Google Places API requests with keyword input to minimise the impact of missing type tags. The used keywords are the ANSI 255 characters, in which each Latin character results in a separate HTTP request. This approach only works at locations with the Latin alphabet as a standard writing, other locations require input from the standard writing for that are. The results of these search will exclude information about the service functions, since the type tags lack, but could offer valuable information about the opening hours of the service.

The S-D search succeeds with radius input that corresponds the double of the bounding sphere radius of the imported 3D data. Several data interpretations functions depend on this double radius in the data combining phase.

5.4.4 Public transportation

The discovery of these transportation nodes results in the generation of TN-D objects instead of BL-D objects. Most transportation nodes lack a referring 3D mesh, causing inaccuracies in the model when BL-D objects store all transportation information. The acquisition of transportation nodes with an illustrative 3D mesh ensue with the public function retrieval as described in the previous paragraph.

The urban analysis tool searches for four type categories in the area: bus station, train station, subway station and transit station. Transit stations describe transportation node points without further specification within the Google database. The TN-D generation process affiliates to the S-D generation process. They differ in the search tags used, as the TN-D generation process only uses the four categories stated above. Also, TN-D objects associate with separate global object lists within the UIM object and have no local connection with any BL-D object whatsoever.

5.5 Combining data

The urban analysis tool combines retrieved data by calculating service function mapping, time of activity and service quantity in the building blocks. Service function mapping calculates the distance between each BL-D object and the nearest BL-D or TN-D object

Category	Google type tags
Education	school, university
Hospitality	bakery, bar, bowling alley, café, casino, food, grocery or supermarket, lodging, meal delivery, meal takeaway, night club, restaurant
Office	accounting, bank, finance, atm, establishment, general contractor, insurance agency, lawyer, moving_company, painter, locksmith, storage
Commercial	amusement park, aquarium, beauty salon, bicycle store, book store, car dealer, car rental, car repair, car wash, clothing store, convenience store, art gallery
Transportation	airport, bus station, gas station, parking, subway station, train station, transit station
Public	city hall, courthouse, church, embassy, fire station, funeral home, hindu temple, library, local government office, mosque, museum, place of worship, police, post office, synagogue
Health	health, pharmacy, physiotherapist, doctor, hospital, veterinary care
Other	park, cemetery

Figure 26: Function type categories

that holds the function of choice. The weighted centre point of all vertices of the BL-D nested mesh marks the reference point of the respective BL-D object. TN-D objects will use the coordinate specified in the object. The interpretation and calculation of the distance needs specification, as multiple definitions of distance exist. For example, the travelled time or obstacle-bypassing path as a result of a A* path-finding algorithm offer vastly different definitions of distance. This version of the urban analysis tool defines distance as the distance as the crow flies, meaning it will calculate the direct ray distance between the two points.

Time of activity uses a global time and weekday filter and compares it to the opening hour data of every S-D object that have been assigned to a BL-D object in the UIM model. A BL-D object that contains at least one open service will be marked as active. The urban analysis tool will waive the calculation of the quantity of activity, as the used API services do not provide in the required information.

Service quantity calculation counts the amount of S-D objects that have been assigned to each BL-D object. This analysis can offer insight in the urban density of the given tissue.

5.6 Data visualization

The urban analysis tool uses the P3D render engine that comes with the Processing 3.0 library. P3D uses OpenGL and suit a broad variety of computer graphic cards. The camera objects offer an isometric view instead of the standard perspective view, so it eases the software in switch between building visualisation and graph visualisation by eliminating the needs to switch between a 3D and 2D render engine. The user can alter the data filters using the control panel in the user interface. The function view mode and distance-to-function view offers a secondary filter option to isolate by function. BL-D objects that do not contain any S-D objects and therefore do not offer function information are drawn black.

The function filters show the product of the function search by type tag analysis. The figure on the right illustrates an example of public services within the urban subject, marked in green. Different type tag filters show different marker colours for the matching BL-D objects. The project site and BL-D entities without S-D objects remain visible in all visualisation filter options. These gaps could theoretically contain a pubic service that was not a part of the Google Maps databases. The figure on the right shows public services with distance mapping. The BL-D objects that contain public services colour purple, the other BL-D objects show their relative distance as the crow flies to the nearest S-D object with a public function. Green resembles a relative short distance, red shows a relative long distance. The time activity filter displays the BL-D objects that contains at least one currently active S-D object. The current time is a global variable for the UIM object and responses to user interface control. The S-D count filter shows to quantity of S-D objects per BL-D object. Green represents a high number of S-D objects, grey a low number. The visualisation can give the architect a good idea where the multi-functional buildings are located in the area. The mesh division process in the data preparation phase can influence the outcome of these analysis.

5.7 Exporting data

The first version of the urban analysis tool offers the possibility to export both view windows as a raster image. It also exports a XLS sheet with an overview of data that has been stored in the UIM object, displaying information about the quantity of objects in the UIM object sorted by service function, opening hours per service and service quantity values in the UIM object. The sheet also shows information about the project site itself, measuring the distance to the closest S-D object of each type.

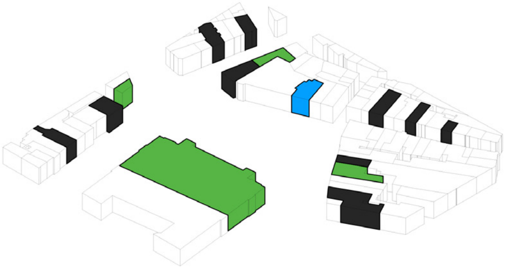


Figure 27: A visualisation sample of the city of Utrecht showing public services in the area

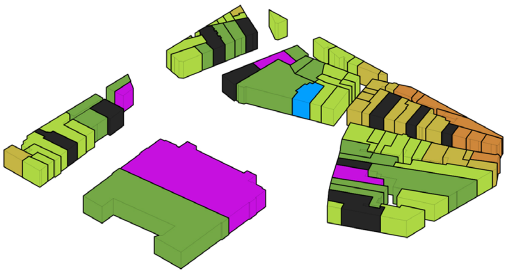


Figure 28: A visualisation sample of the city of Utrecht showing public services and distance mapping towards those services

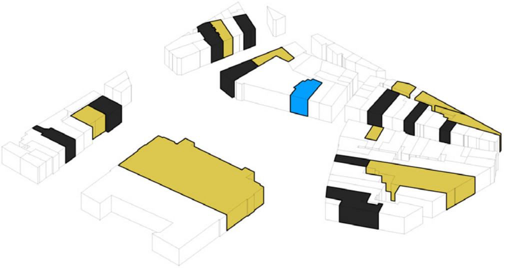


Figure 29: A visualisation sample of the city of Utrecht showing building blocks with activity at current time

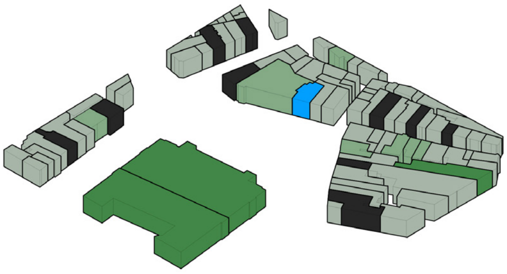


Figure 30: A visualisation sample of the city of Utrecht showing the quantity of services per building block

6.
CASE STUDIES

RATHAUSSTRASSE. 1, VIENNA, AUSTRIA

6.1.1 General project information

The building site of the first case study is situated at Rathausstraße 1 in Vienna, Austria. In 2013, a call for architectural design of an office building for the City of Vienna has been made public. The architectural competition has been won in the second phase by a collective of Stadler Prenn Architekten GbR, Schuberth und Schuberth ZT-KG and Ostertag Architects. The building site is situated in the 1st district of Vienna, making it a fitting subject for the first case studies. The direct environment of the building site contains multiple buildings of high municipal and national value, like the city hall of Vienna, the historic Burgtheater, the main building of the University of Vienna and the National Parliament of the Republic of Austria. For this case study, we will pretend as if there has not been a winning project yet. Therefore, we can approach the urban analysis work-flow with the same amount of information as a regular candidate office would have for this architectural competition.

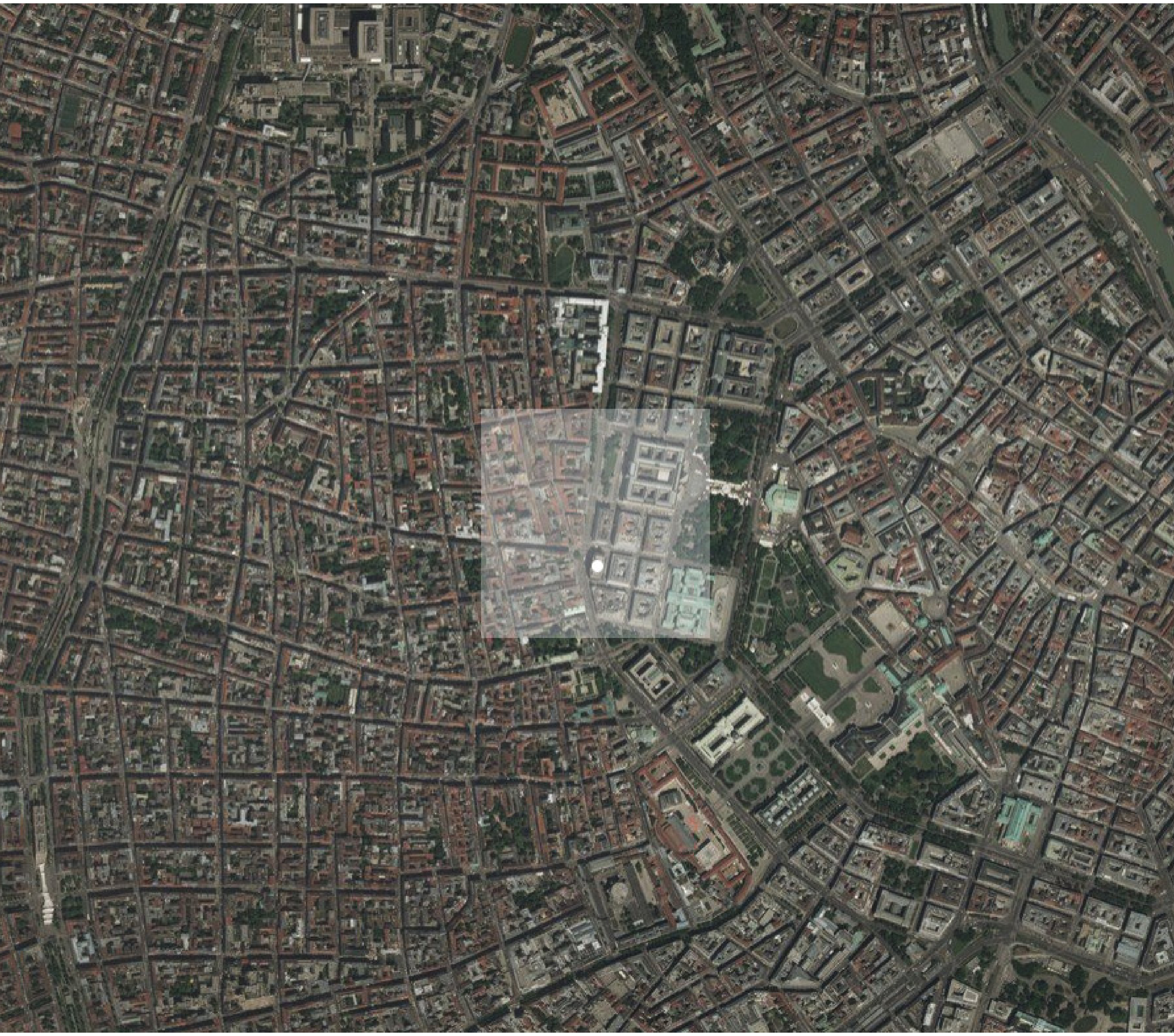


Figure 31: The area around Rathausstraße 1

6.1.2 Pre-existing 3D data

The city of Vienna offers free geometric data of the region within their municipal borders. The available data types are 3D building data, 3D landscape data, satellite images, Digital Surface Models (DSM) and 2D maps. The urban analysis tool only requires 3D building data at a minimum of LOD 0 detail level. Therefore, the 3D building data at LOD 2 level in the area of Rathausstraße 1 within a radius of 250m will be sufficient for the case study and manually downloaded in DXF format.

Once imported in Rhinoceros 5, the 3D data will look like the picture on the right. The data does not need to be filtered for unnecessary object, because the model does not contain additional guidelines, landscape objects or null objects. The bottom surfaces of the volumes are open. They do not need additional editing, as it aspect will not affect the analysis functions of the urban analysis tool.

The centre point of the model is the only aspect that requires adjustments. Currently, the centre point of the model holds a coordinate that deviates 341.15 meters to the south of the actual 3D data. This will result in camera problems in the analysis tool. Therefore, the 3D data will be translated for this distance in the negative direction of the Y-axis. Now the centre point of the 3D model at {0,0,0} corresponds with the average value of its vertexes. The scale of the model is already 1:1 and does not to be corrected.

The building volumes already have been separated in different meshes, making it easy to export the 3D data to separate STL documents. The export process have been performed with a custom written export script in the Python code language, resulting in an export folder with 489 unique STL files.

The local-global coordinate link connects local coordinate {0,0} to global coordinate {48.211774, 16.356292}. The mesh with ID number 67 resembles Rathausstraße 1 and has will mark the building site.

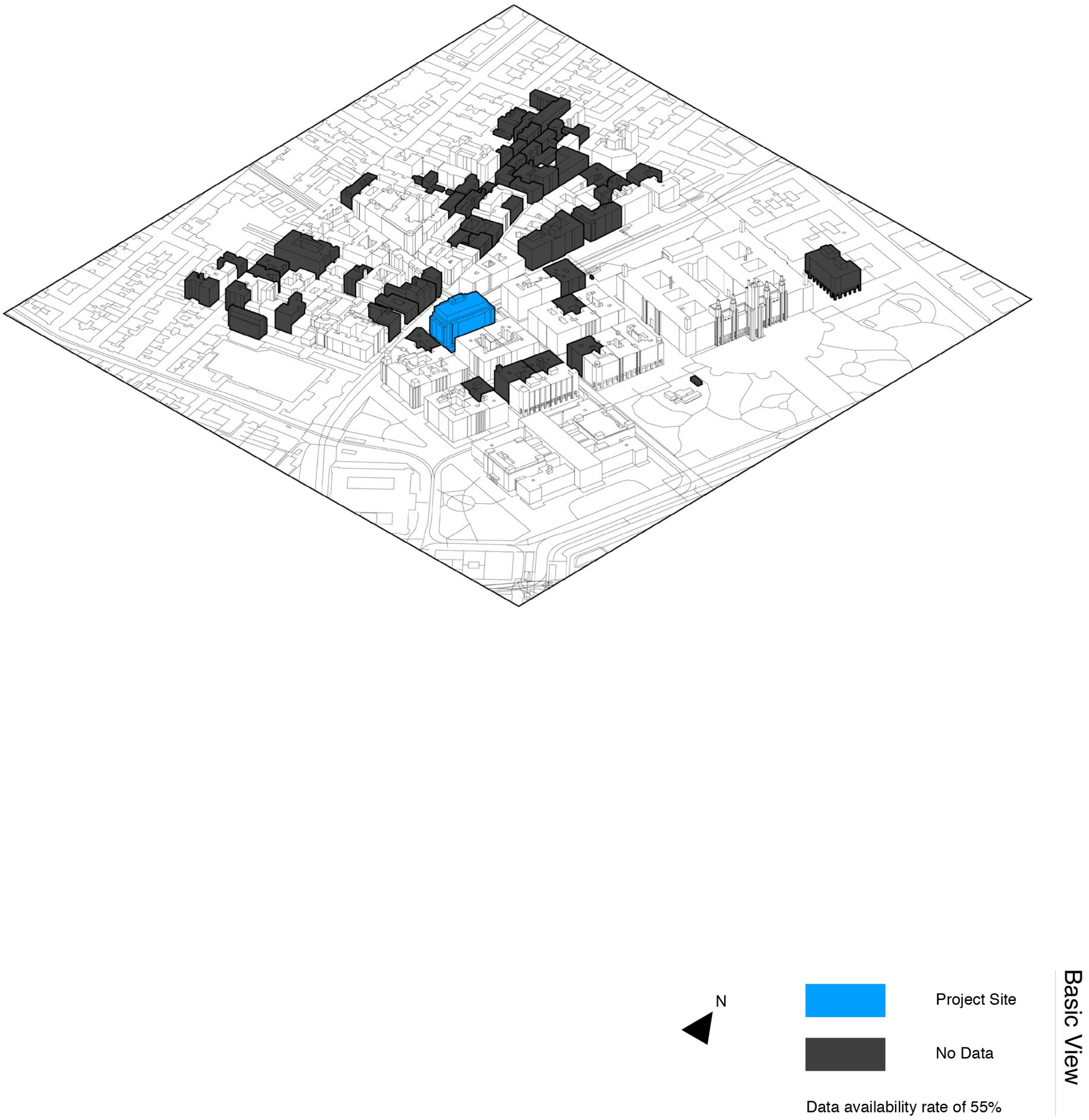


Figure 32: The imported 3D building data of the area around Rathausstraße 1

6.1.3 Results

The area only contains one building with educational services, at the Friedrich-Schmidt Platz 5. This address contains multiple institutes aimed at youth support, among them WienXtra and Wiener Jugendkreis. Hospitality is widely spread over the area, with a high density of catering related service along the Josephstädterstraße, an important connection to the Rathaus Square and the Gürtel axis in the west. This high density zone is right next to the project site. Office spaces are also wide spread across the area. There is no clear pattern for the office function. The area has a low quantity of commercial services. The city hall can be found in the north-east. A city office can be found in the adjacent building in the east of the project site. Health functions are also widely spread across the area. For most of the services it is questionable if they are really health-related, since these are beauty or massage salons that have marked themselves health-related. The metro station Rathausplatz is located north of the project site. There are several transit stations located at the Ringstraße, which is located at the east of the site.

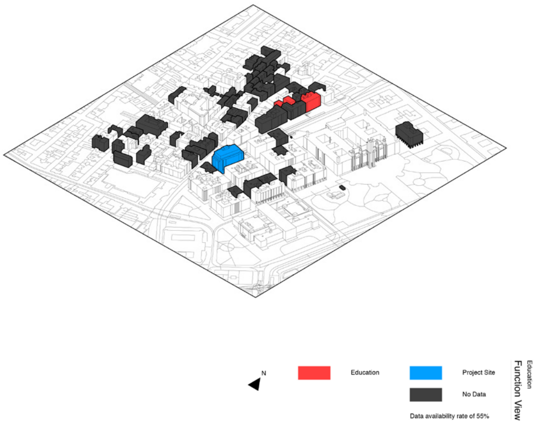


Figure 33: Education function view

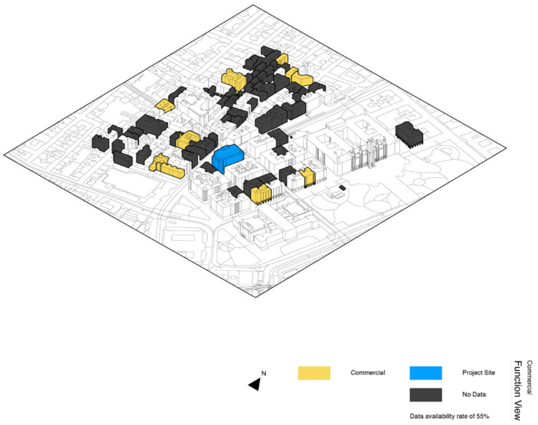


Figure 36: Commercial function view

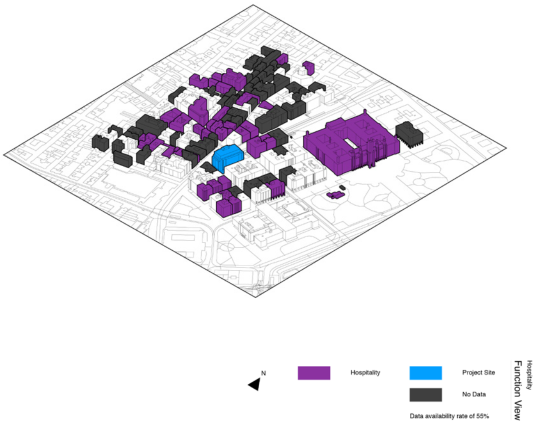


Figure 34: Hospitality function view

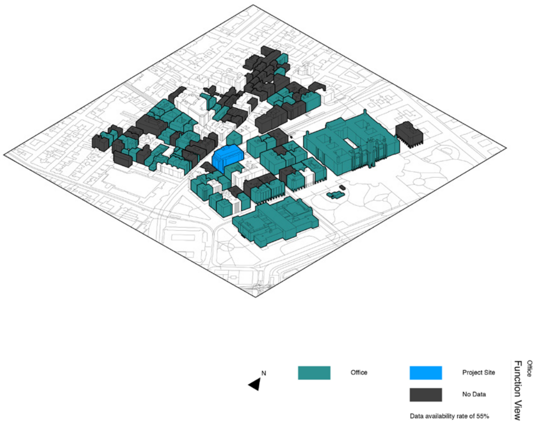


Figure 35: Office function view

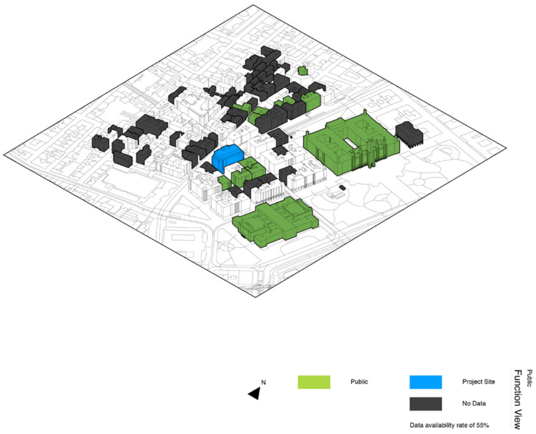


Figure 37: Public function view

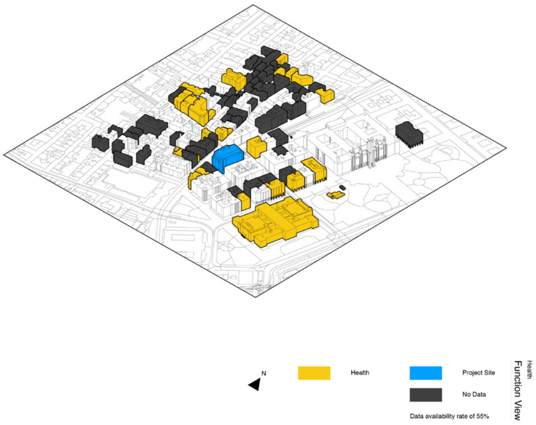


Figure 38: Health function view

Related education functions are located within zones that are farther from the open Ringstraße in the east. Hospitality functions are even spread across the area. There is also no clear zone pattern for office functions. Commercial services are even spread across the area as well. The Landesgerichtsstraße, directly west of the project site, is an important axis for public transportation in the area. Public functions are orientated towards the Ringstraße in the east. Health functions are orientated towards the Ringstraße in the east.

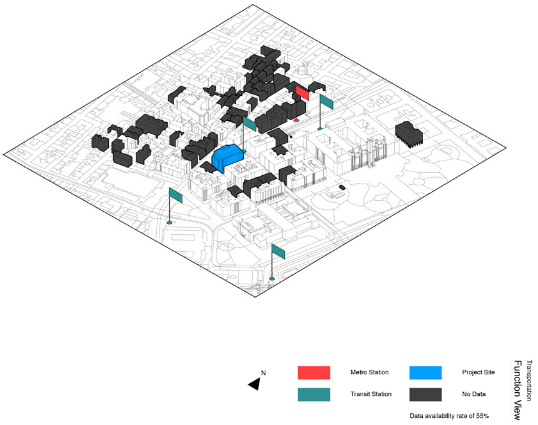


Figure 39: Transportation function view

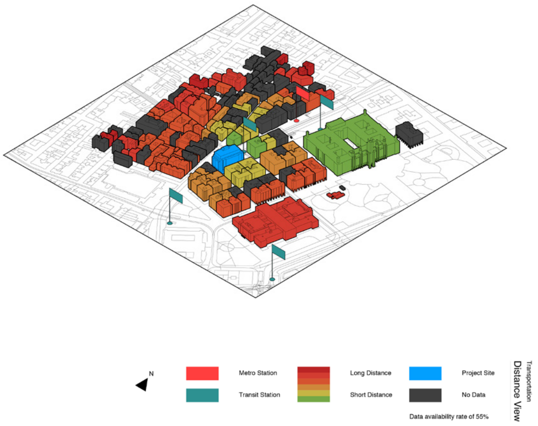


Figure 42: Transportation distance view

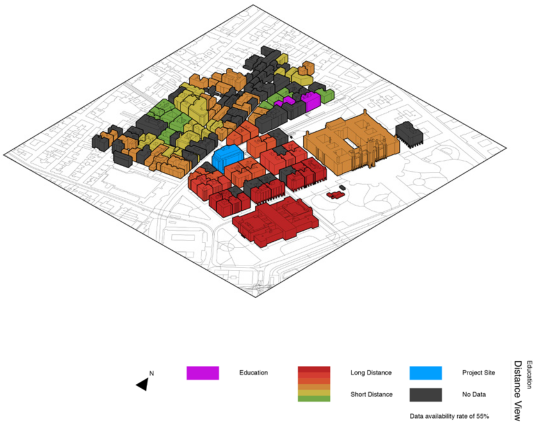


Figure 40: Education distance view

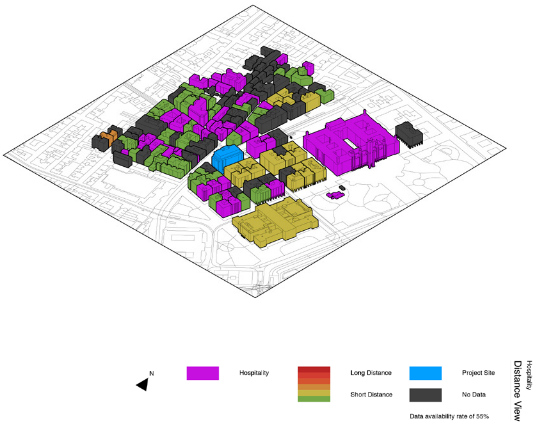


Figure 41: Hospitality distance view

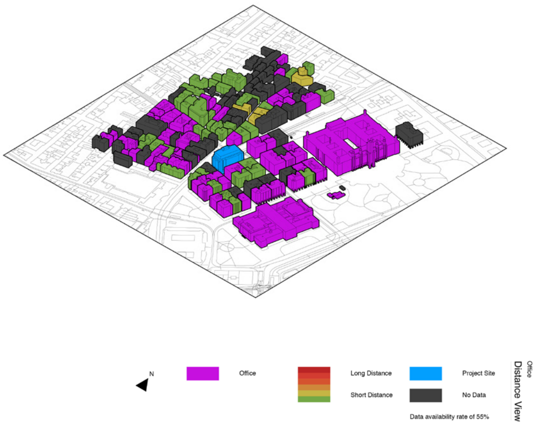


Figure 43: Office distance view

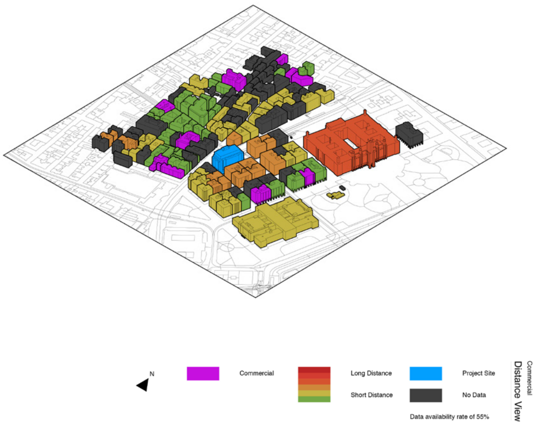


Figure 44: Commercial distance view

Most buildings show a low quantity of unique services within their shell. The only remarkable building volume is the city hall building on the right side of the picture.

There is only small activity in the early morning at the Stadiongasse, north of the project site. These buildings contain several bank offices. Almost all functions are open at midday. Almost one half of the services have been closed by the early evening. The building with the bank offices at Stadiongasse 10 is still active.

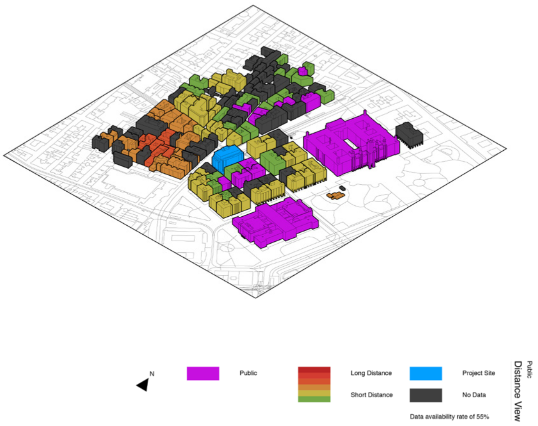


Figure 45: Public distance view

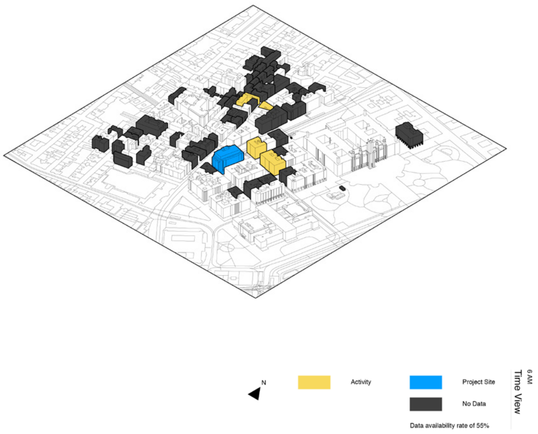


Figure 48: Time view 6AM

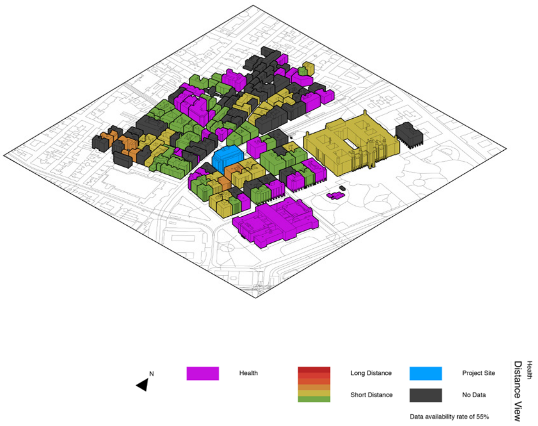


Figure 46: Health distance view

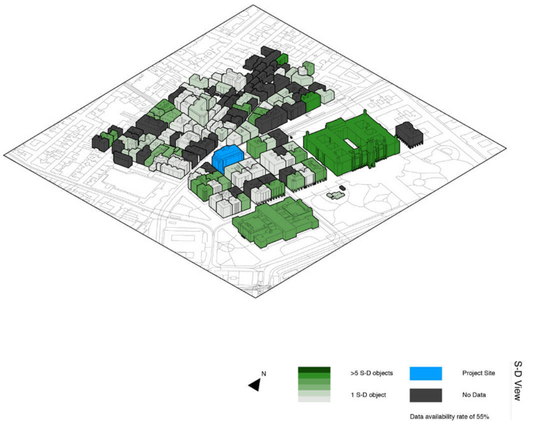


Figure 47: S-D quantity view

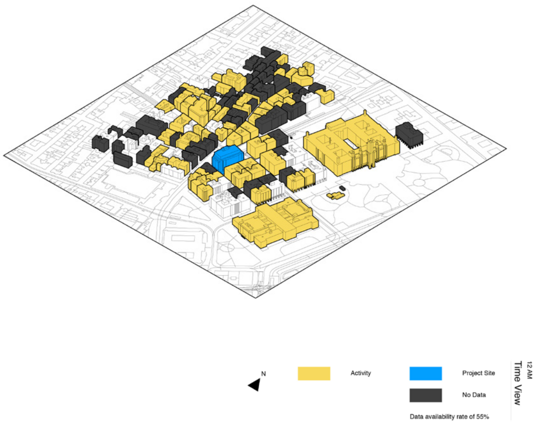


Figure 49: Time view 12AM

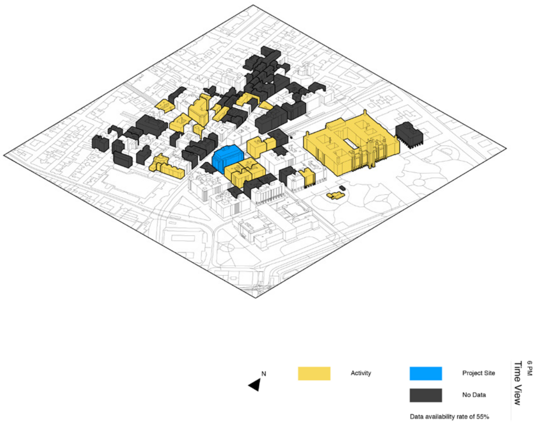


Figure 50: Time view 6PM

The figure on the right shows service functions that can be found in the UIM model. The area around the Rathausstraße contains a high percentage of office, public and hospitality functions.

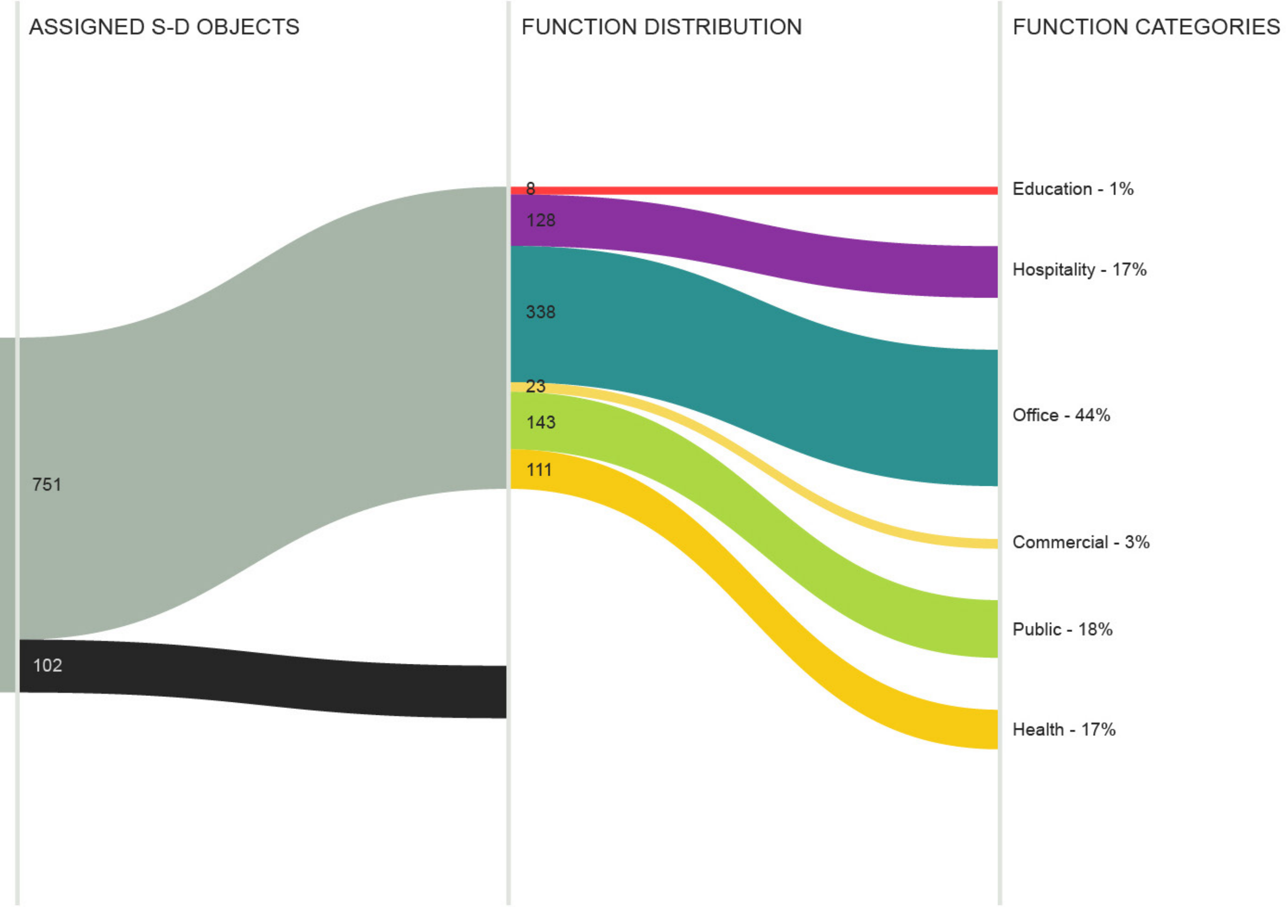


Figure 51: Division of service functions

CHARTERHOUSE SQUARE
LONDON, UNITED KINGDOM

6.2.1 General project information

Charterhouse Square is located in Central London. The square is not subject of an existing architectural competition, like the case study at the Rathausstraße in Vienna, but its surrounding are still an interesting topic of analysis. Both locations offer similarities in their available services (both areas have important academic infrastructure nearby and contain attractive tourist spots), but both cities differ vast in scale, history and architectural roots. Since there is no reference of a recent architectural competition in the direct area, there is full freedom to choose the building site. The building site for the second case study is therefore one of its most interesting buildings: The Queen Mary University of London main building, located at the Charterhouse Square Campus.

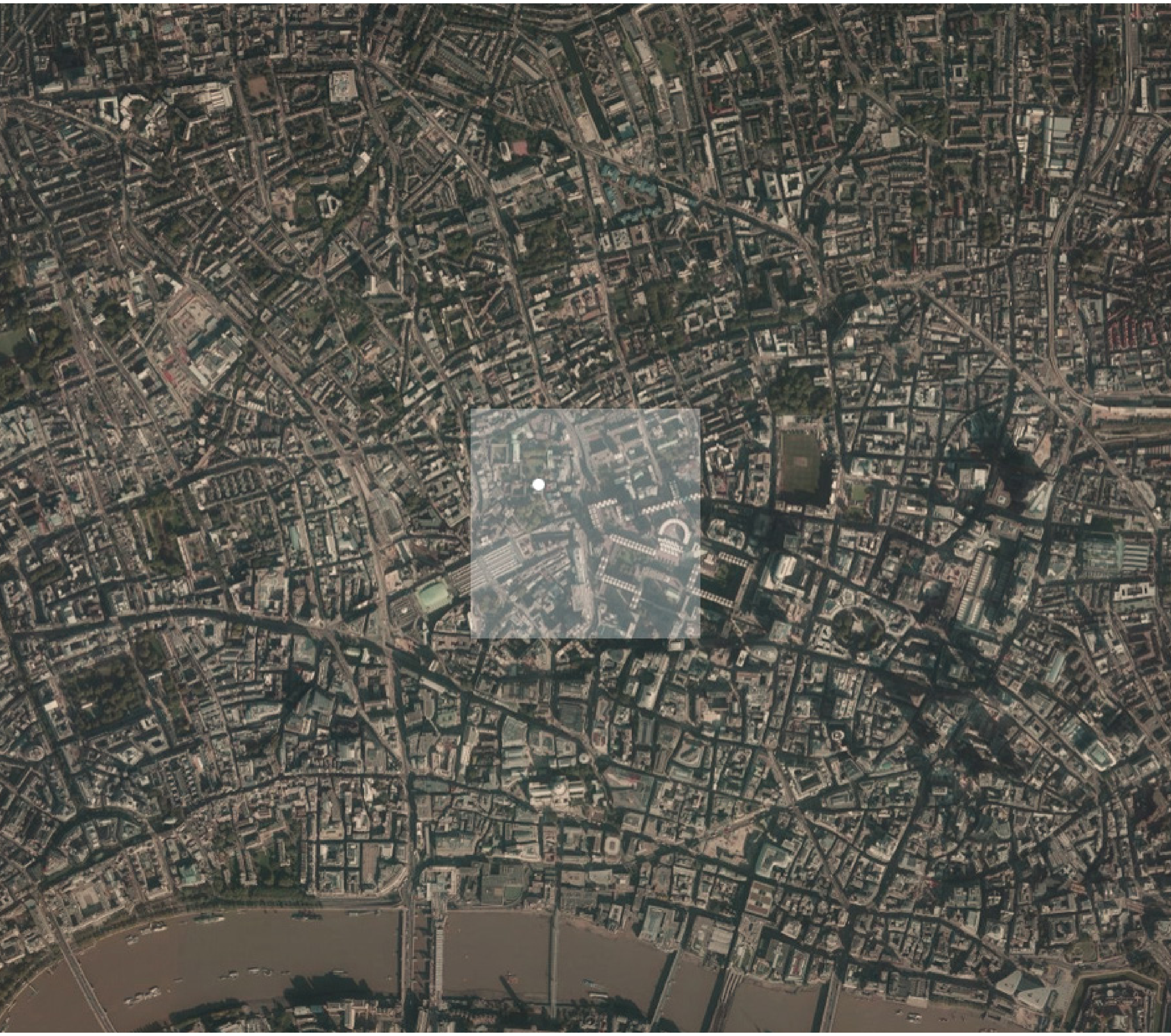


Figure 52: The area around Charterhouse Square

6.2.2 Preparing pre-existing 3D data

The City of London does not offer 3D data about her municipal region as open and free as the City of Vienna. Nevertheless, there are several urban 3D companies in London that offer sample data from their databases. The 3D model of the Charterhouse Square in London has been produced by the British company “Vertex Modelling” and is offered for free to their future clients in a range of LOD 1 to LOD 4. This case study makes use of the data at LOD 2 level, due its good balance between building detail and polygon count.

The weighted centre of the 3D data already matches the centre point of the local axis system and does not to be adjusted. The document does contain objects for landscape elements and vegetation and need to be filtered out. In contrast to source material for the first case study, this data does not contain separate meshes for each building block. These large meshes will influence the results of the urban analysis tool in a negative way and need to be sliced. Each mesh is sliced in a block of 5x5 meter.

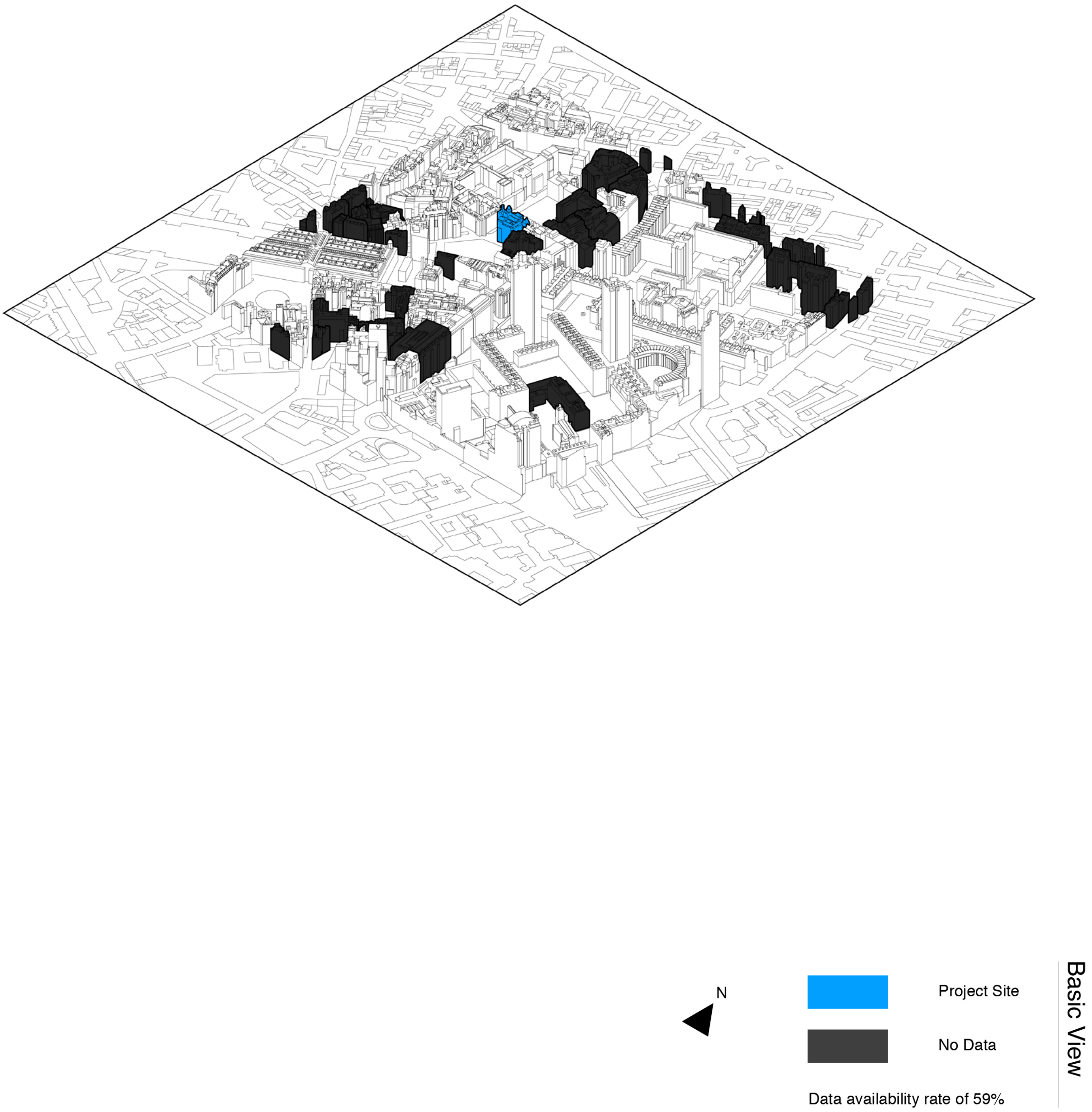


Figure 53: The imported 3D building data of the area around the Charterhouse Square

6.2.3 Results

The project site lies next to the Charterhouse Square Campus. The other buildings contain educational services for the neighbourhood, among them a childcare and a primary school. There is a wide spread of hospitality services with a high density. The office services show a similar pattern as the hospitality functions. The tower at Lauderdale Place 1 contains a computer store. The commercial surrounding services differ in specific function. The public services in the image are all religion related. Charterhouse Chapel is located at Charterhouse Square 14, south of the project site. The London City Presbyterian Church is located central south, at Aldergate street 15. There is a wide spread of small private health institutions in the south of the area. Barbarican Metro Station is located south-east of the project site. There is a high spread of public transit stations across the area.

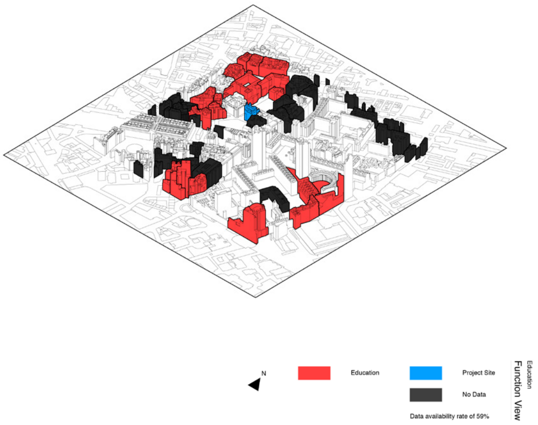


Figure 54: Education function view

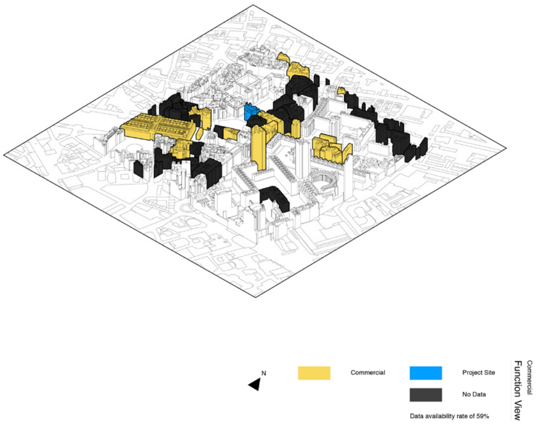


Figure 57: Commercial function view

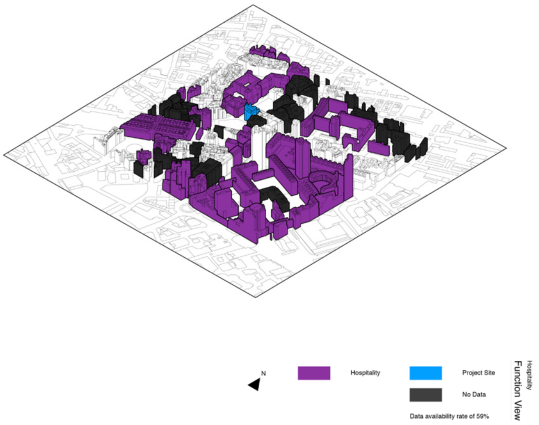


Figure 55: Hospitality function view

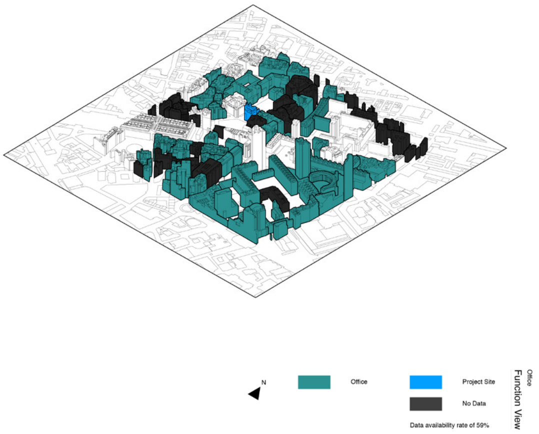


Figure 56: Office function view

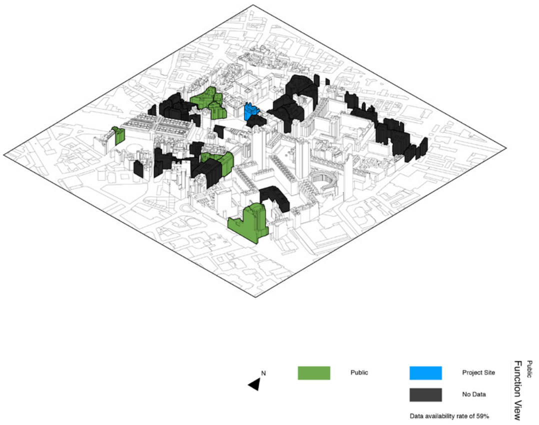


Figure 58: Public function view

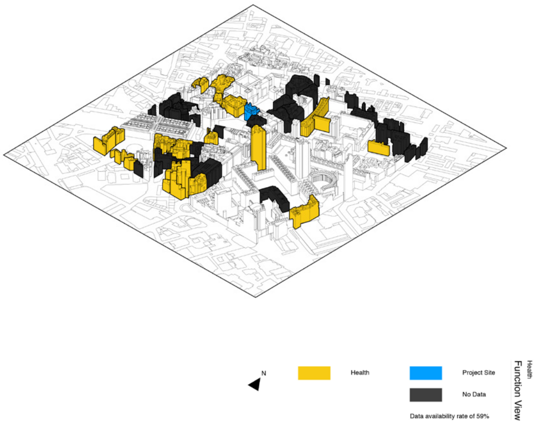


Figure 59: Health function view

Education services are located at the outside of the area. There is a strong density of hospitality services in the south-east of the area, in the direction of the museum of London. There is also a clear zone of office functions in the south-east of the map, but this zone draws further over the Charterhouse Square Campus and the project site. Commercial functions are situated around the tower at Lauderdale Place 1. There is a wide spread of public transit station in the area, but there is a strong density of stations visible at the Goswell road. This axis lies east of the project side and is drawn north to south. Public functions are even spread across the area. Health functions are also even spread across the area.

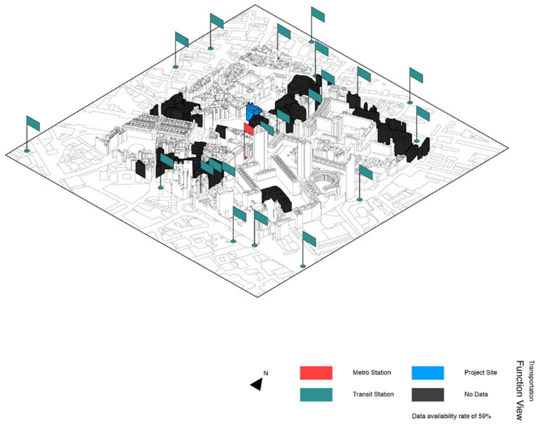


Figure 60: Transportation function view

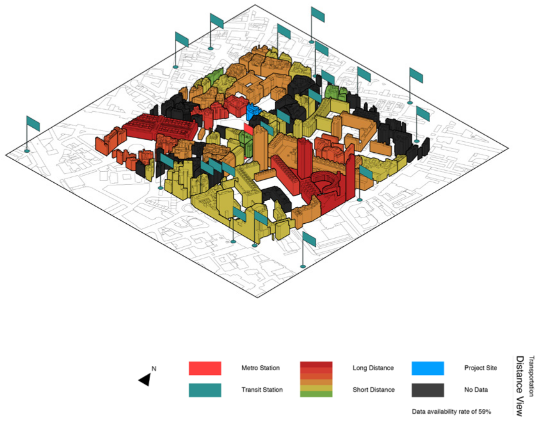


Figure 63: Transportation distance view

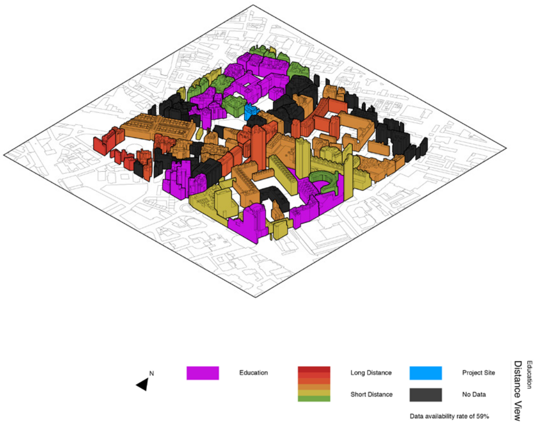


Figure 61: Education distance view

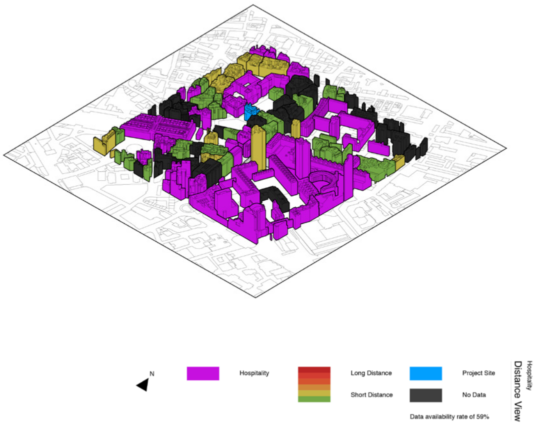


Figure 62: Hospitality distance view

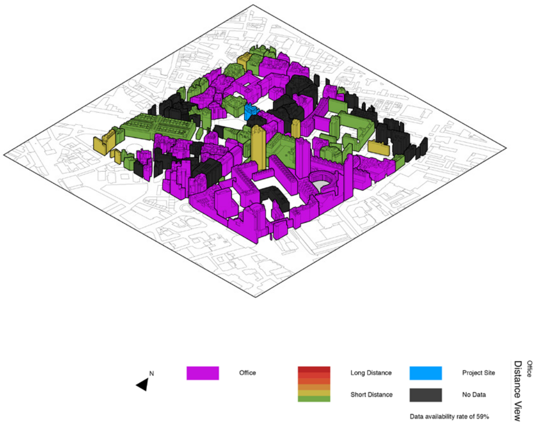


Figure 64: Office distance view

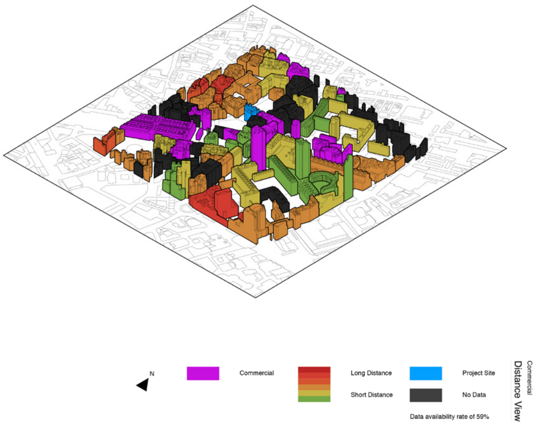


Figure 65: Commercial distance view

The area shows several buildings with a high quantity of services, but the most remarkable is the building in the south at Aldergate street 200. This building is a multi-functional office building that is draped around the Museum of London.

There is almost no service open in the area early in the morning. At midday, one half of the services has opened. The image of the early evening shows barely any difference from midday.

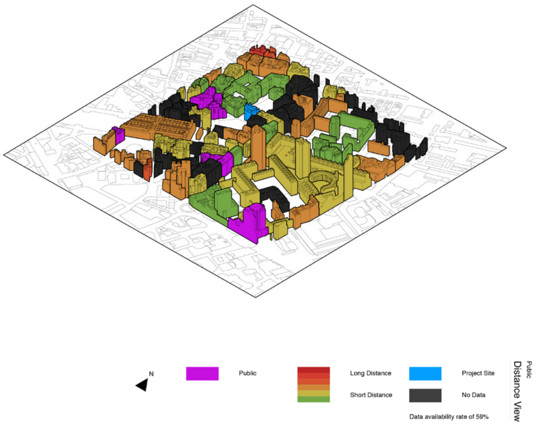


Figure 66: Public distance view

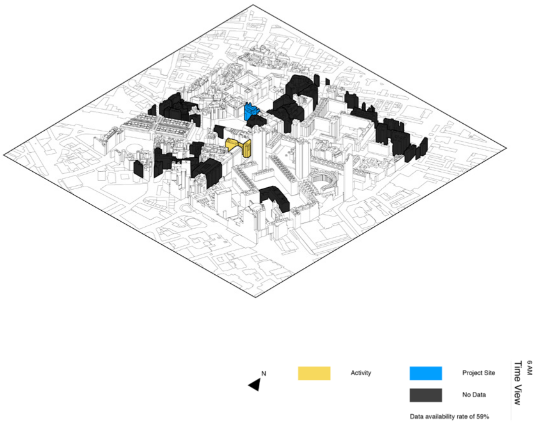


Figure 69: Time view 6AM

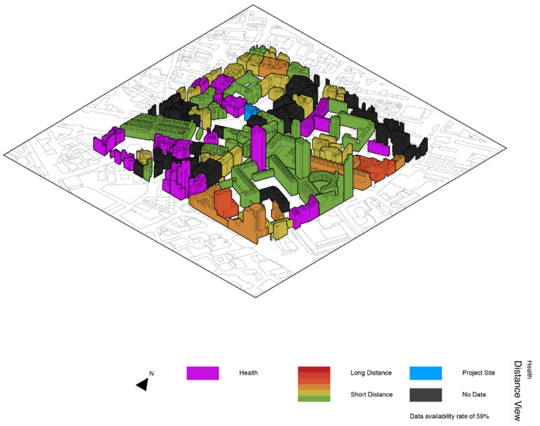


Figure 67: Health distance view

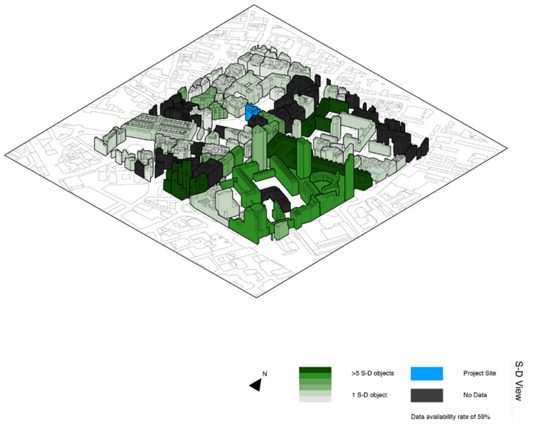


Figure 68: S-D quantity view

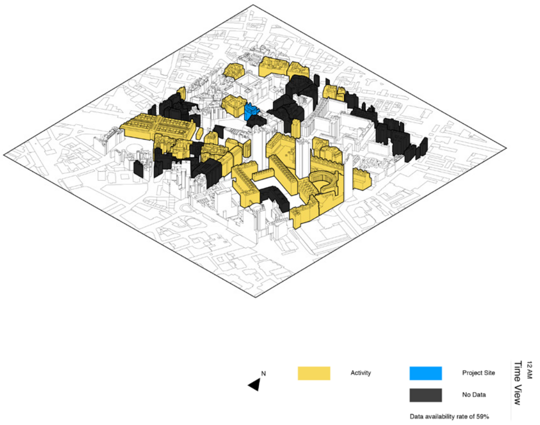


Figure 70: Time view 12AM

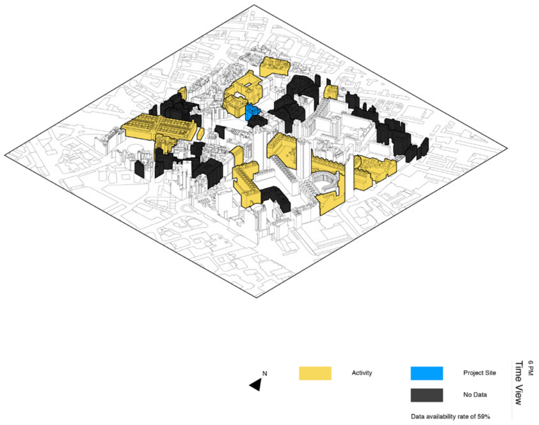
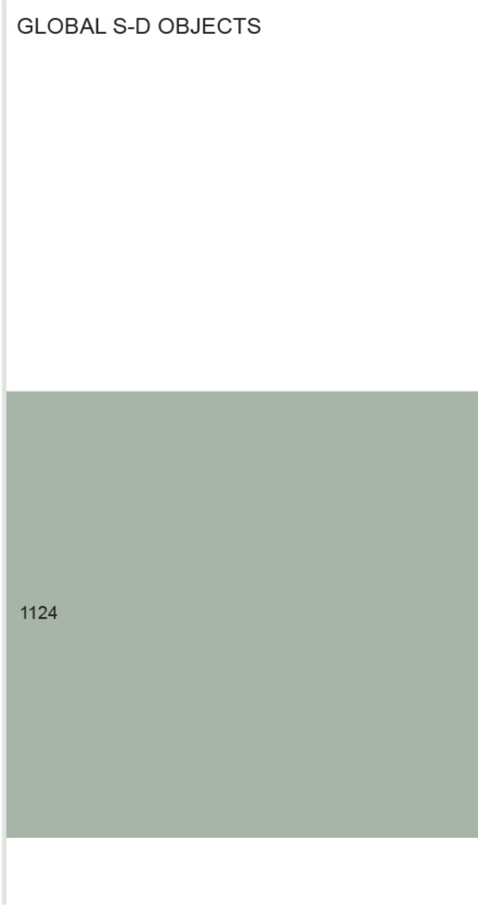


Figure 71: Time view 6PM

The figure on the right shows service functions that can be found in the UIM model. The area around the Charterhouse Square contains a high percentage of office and hospitality functions.



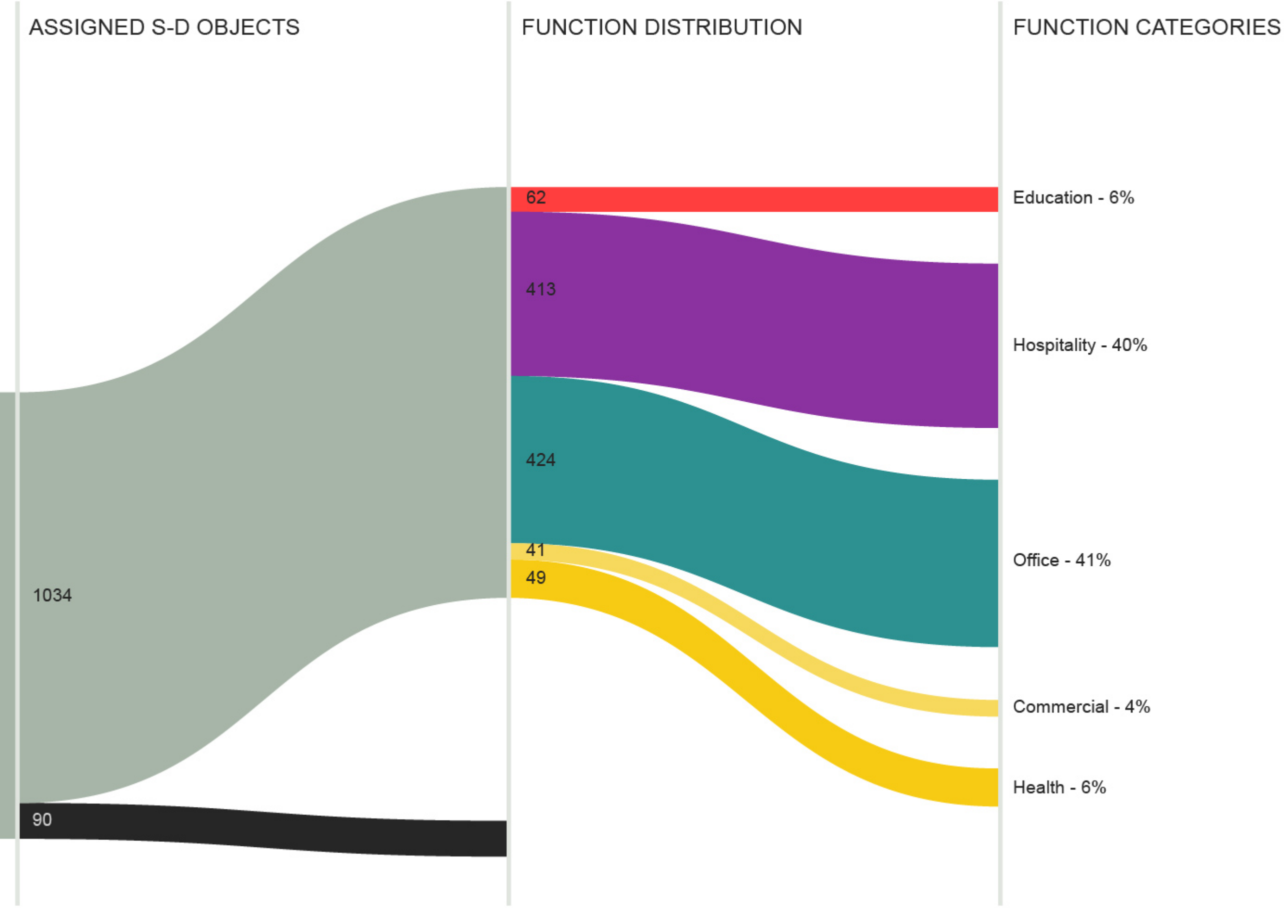


Figure 71: Division of service functions

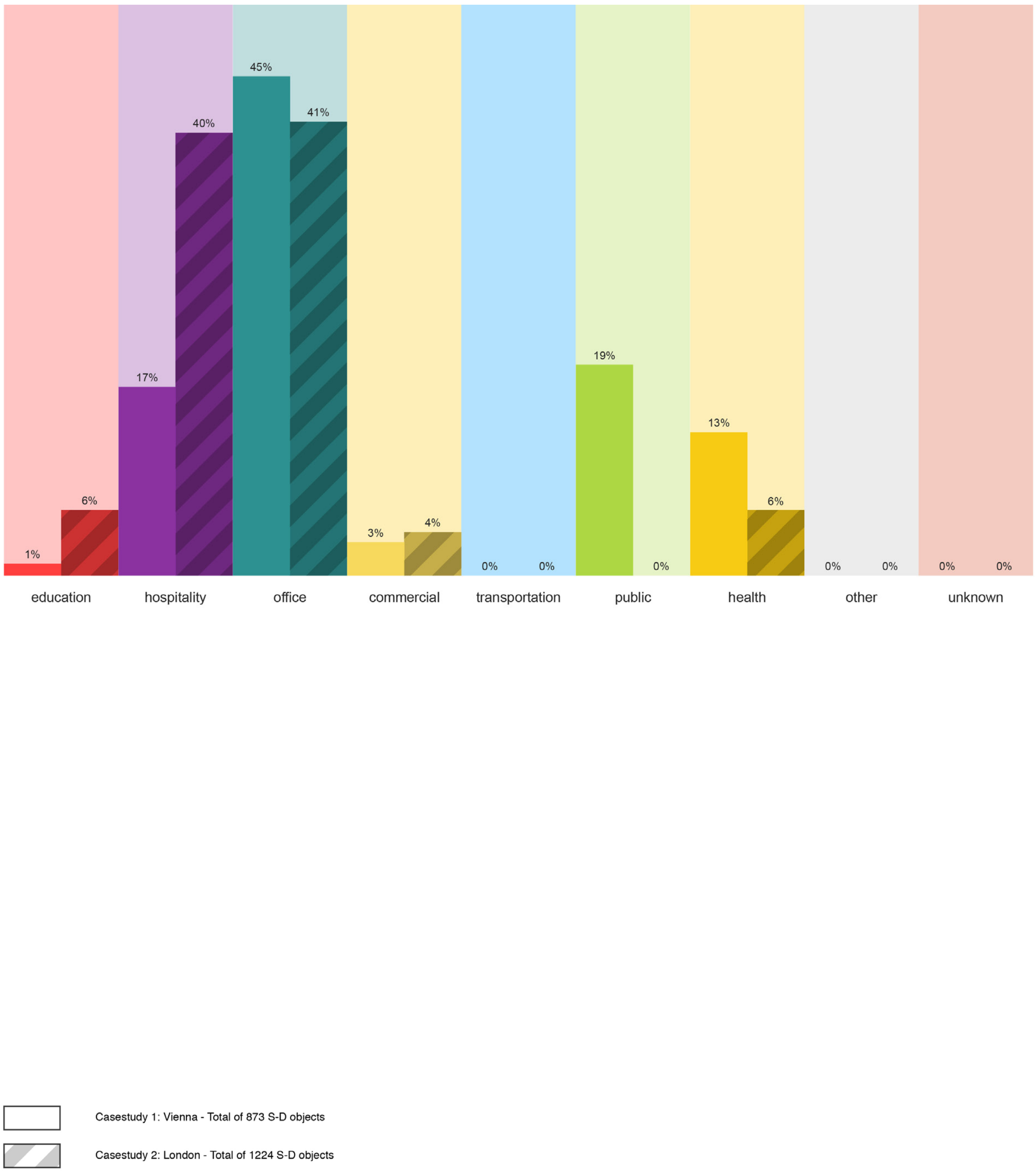
6.3 Comparing the case studies

The figure on the right shows that the Vienna case study subject contains a relative higher quantity of public functions and office. This could be explained by the fact that the city hall of Vienna and the national parliament are located near the project site. The area around the Charterhouse Square in London shows an even distribution of office and hospitality functions.

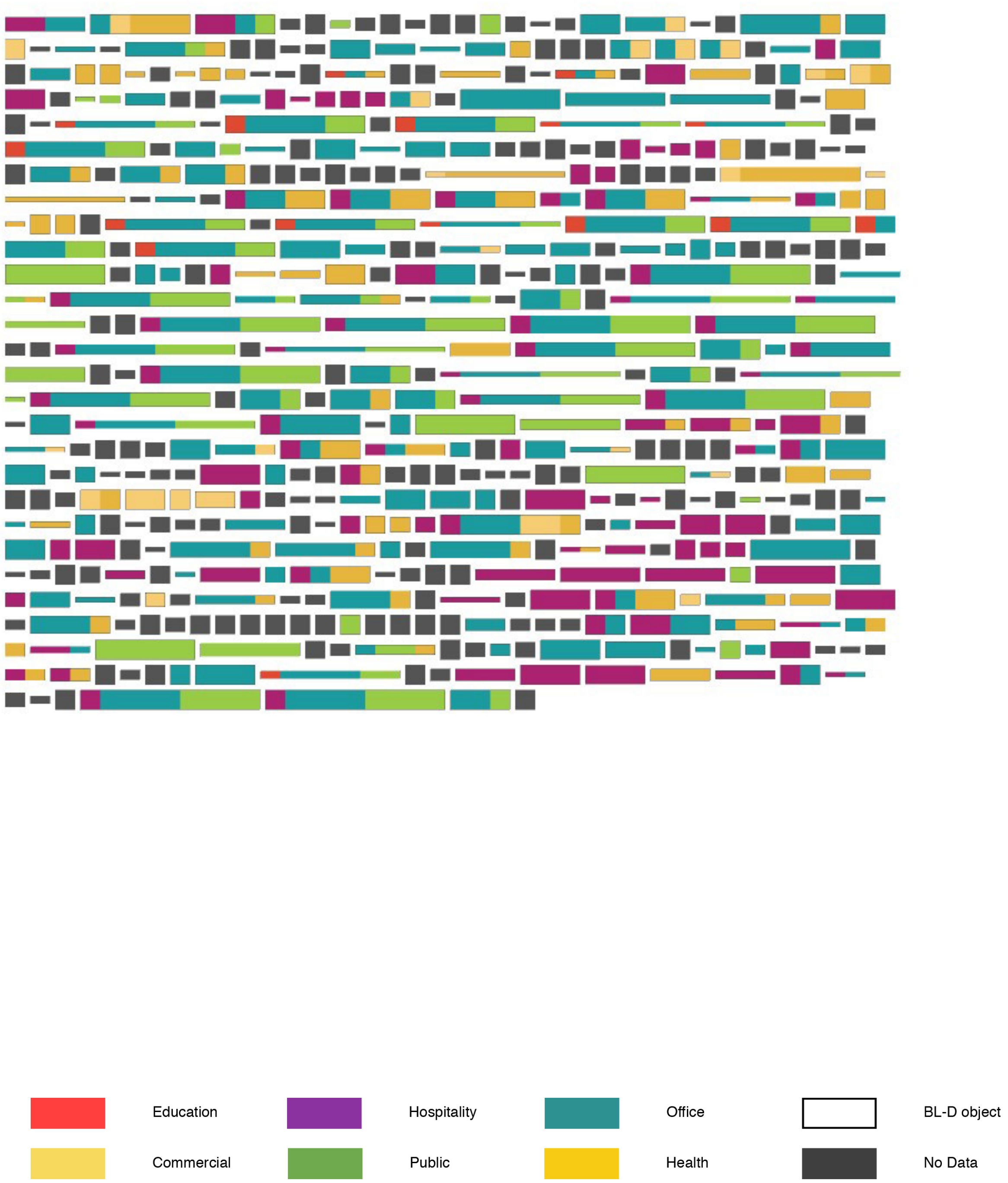
The categories only show function above a specific quantity threshold. The zero percent markers in the graph indicate that the urban tissue contains less items of the given category than this number. The quantity threshold for the Vienna case study is 8, the quantity threshold for the London case study is 12. Any category that contains less items is displayed as zero percent.

The figures on the next spread show that the Vienna case study subject contains a relative higher quantity of public functions and office. This could be explained by the fact that the city hall of Vienna and the national parliament are located near the project site. The area around the Charterhouse Square in London shows an even distribution of office and hospitality functions.

The categories only show function above a specific quantity threshold. The zero percent markers in the graph indicate that the urban tissue contains less items of the given category than this number. The quantity threshold for the Vienna case study is 8, the quantity threshold for the London case study is 12. Any category that contains less items is displayed as zero percent.



45%
41%
40%
19%
17%
13%
6%
6%
4%
3%
1%
0%
0%
0%
0%
0%
0%
0%
education
hospitality
office
commercial
transportation
public
health
other
unknown
□ Casestudy 1: Vienna - Total of 873 S-D objects
▨ Casestudy 2: London - Total of 1224 S-D objects



String height represents relative mesh volume of parent BL-D object

Figure 73: Division of service functions in the area around Rathausstraße 1 within each BL-D object

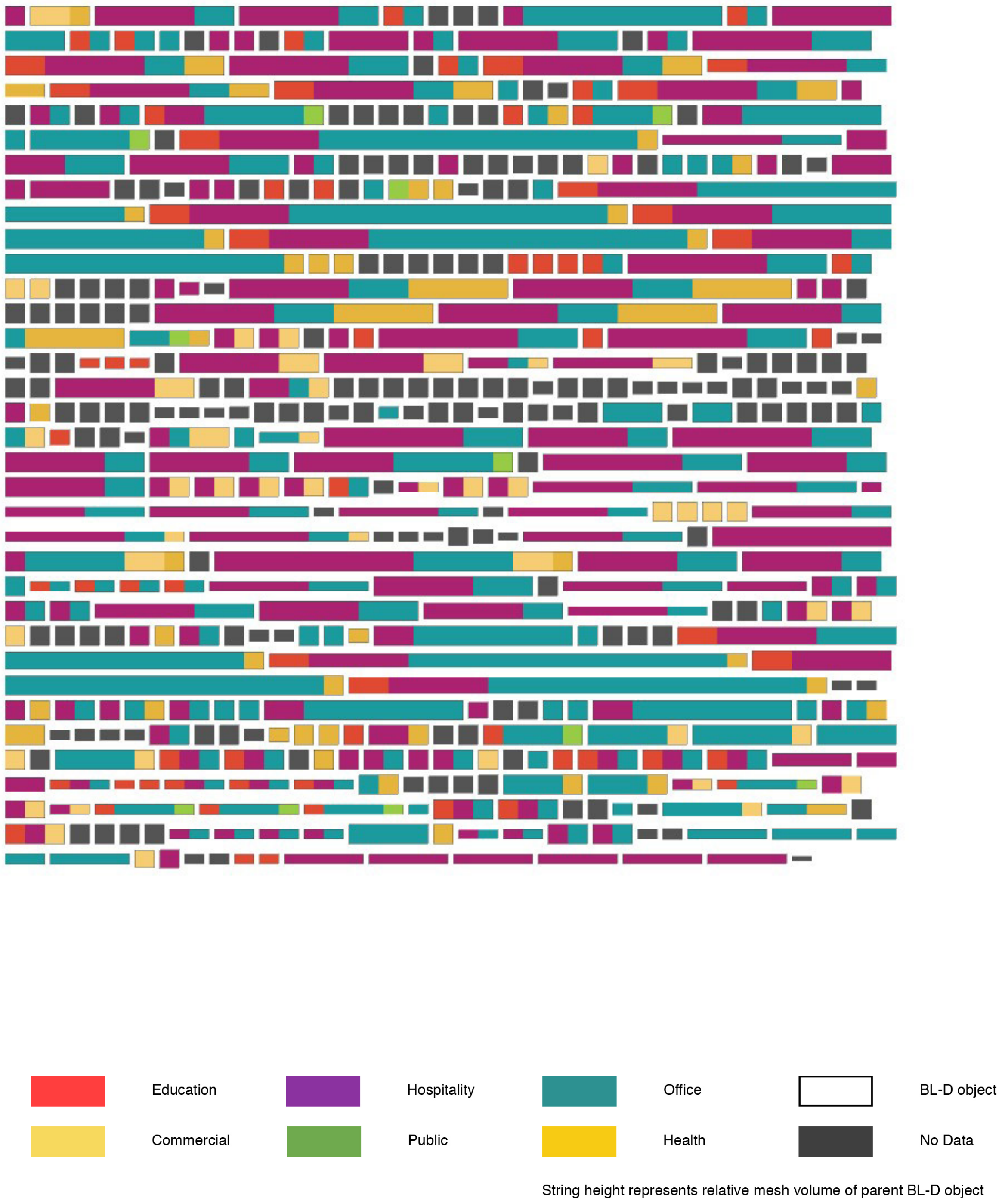


Figure 74: Division of service functions in the area around the Charterhouse Square within each BL-D object

7.
DISCUSSION

7.1 Reflection on final product

The urban analysis tool can serve as a handy tool to gain quick and basic insight of the patterns in the urban tissue. Though there are still some issues that can cause inaccuracies, making it a necessity for the user to perform further research on interesting patterns that have been found during the use of the tool. The tool only uses the Google Maps API as its main source of information. The stored data from this database relies on the input of its users, consisting of commercial parties, public institutes among others. This factor increases the chance of overlooked services or inaccurate service descriptions.

7.2 Future development and improvement

The prototype of the urban analysis tool, as presented in this thesis work, could be improved on multiple facets. Improving the accuracy of the current implemented analysis could be a great starting point. This can be achieved by using retrieving information from different online databases, then comparing the information from both sources and complement missing elements. A good choice for a second data source would be one that doesn't rely on the input of its users, but has been filled out by its supervising organisation. Municipal or national archives would meet this requirements, though these databases rarely come with an API. The user should download this data manually and load it into the urban analysis tool as additional pre-existing data.

Secondly, the tool could be expanded with additional analysis functions. Interesting topics for analysis could be spatial qualities, landscape properties, flora, legal conditions, historic value, monumental state, climate and property pricing.

Also, there is still space for improvement on the connection between the urban analysis tool and the digital tools that the architect uses for the development of the conceptual design. The current version of tool cannot export its information in 2D vector maps yet. An addition of these export modules would benefit the integration of the analysis tool and 2D CAD software. An interesting alternative is redeveloping the urban analysis tool as an integrated plug-in for CAD software.

7.3 Possible applications

The urban analysis tool could offer great value in the setting of the architectural competition: it reduces the time required for simple urban analysis and can makes complex, time-consuming analysis attainable in project with short development times. Of course, these benefits also apply to regular architectural projects. The tool can be used for the development of a spatial program for the given site or test multiples site for their fitness with a fixated architectural program. The harvested data is absolute and is therefore fit for use in digital processes that require absolute information. This could span from the process of generating or parametrizing architectural design to AI systems that benefit from understanding the urban environment.

REFERENCES

Bibliography

Batty M, Dodge M, Jang B, Smith A. 1998 *“GIS and Urban Design*” Center for Advanced Spatial Analysis – Working Paper Series

Beirao J, Duarte J, Stouffs R. 2007 “*Structuring a Generative Model for Urban Design: Linking GIS to Shape Grammars”* eCAAD 26 – Section 23: City Modeling 2

Beirao J, Duarte J. 2002 *“Urban Grammars: Towards Flexible Urban Design”* eCAAD 23 – Section 11: Shape Grammars

Buuksahlih I, Isikdag U, Zltanova S. 2013 *“Exploring the Processes of Generating LOD (0-2) CityGML Models in Greater Municipality of Istanbul”* International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XL-2/W2

Fang C, Yichun X. 2008 *“Site Planning and guiding principles of hi-tech parks in China: Shenzen as a case study.”* Environment and Planning B Volume 35, pages 100-121

Fischer A, Kolbe T, Lang F. 1998 *“Extracting Buildings from Aerial Images Using Hierarchical Aggregation in 2D and 3D”* Computer Vision and Image Understanding, Volume 72, pages 185-203

Hutson A. 2004 *“The Urban Design Machine”* The 38th International Conference of Architectural Science Association

Keim D. 1997 *“Visual Data Mining”* International Conference on Very Large Databases (VLDB’97)

Liggett R, Jepson W. 1994 *“An integrated environment for urban simulation”* Environment and Planning B: Planning and design 1995, volume 22, pages 291-302

Lynch K, Hack G. 1974, *“Site Planning, Third Edition”* The MIT Press

Mayer H. 1999 *“Automated Object Extraction from Aerial Imagery – A Survey Focusing on Buildings”* Computer Vision and Image Understanding, Volume 74, pages 138-149

Pollefeys M., Nister D., Frahm J.. 2008 *“Detailed Real-Time Urban 3D Reconstruction from Video.”* Int J Comput Vis, Volume 78, pages 143-167

Suveg I, Vosselman G. 2004 *“Reconstruction of 3D building models from aerial images and maps”* Journal of Photogrammetry & Remote Sensing, Volume 58, pages 202-224

Zhou G, Song C, Simmers J. 2004 *“Urban 3D GIS From LiDAR and digital aerial images”* Computers & Geosciences, Volume 30, pages 345-353

https://www.wien.gv.at/ma41datenviewer/public/ (26-05-2017)

Figures

[1] Batty M, Dodge M, Jang B, Smith A. 1998 *“GIS and Urban Design”* Center for Advanced Spatial Analysis – Working Paper Series, p. 7

[2] http://www.oasys-software.com/media/Agent%20Time%20until%20clear.png (08-07-2017)

[3] http://cloud.originlab.com/images/WikiWeb/Graphing/3D_Scatter_with_Colormap.png?v=3627 (14-06-2017)

[4] http://www.visualisingdata.com/blog/wp-content/uploads/2012/07/TDFColours-600x402.png (14-06-2017)

[5] http://arcadenw.org/images/uploads/content-media/UNAIDS_Treemap_full.jpg (14-06-2017)

[6] http://www.blankchapters.com/wp-content/uploads/2013/03/black-background.png (14-06-2017)

[7] Liggett R, Jepson W. 1994 *“An integrated environment for urban simulation”* Environment and Planning B: Planning and design 1995, volume 22, p. 292

[8] Liggett R, Jepson W. 1994 *“An integrated environment for urban simulation”* Environment and Planning B: Planning and design 1995, volume 22, p. 294

[9] Beirao J, Duarte J, Stouffs R. 2007 *“Structuring a Generative Model for Urban Design: Linking GIS to Shape Grammars”* eCAAD 26 – Section 23: City Modeling 2, p. 930

[10] http://www.simstadt.eu/design/images/LODs.png (14-06-2017)

[11] Buuksahlih I, Isikdag U, Zltanova S. 2013 *“Exploring the Processes of Generating LOD (0-2) CityGML Models in Greater Municipality of Istanbul”* International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XL-2/W2, p. 22

[12] https://www.netwasgroup.us/services-2/images/202_75_136-latitude-and-longitude-point-globe.jpg (14-06-2017)

A.

CODE SNIPPETS

```
//Retrieve JSON document using HTTP request

public JSONObject httpRequest(String APIlink ){

        //Initialise variables
        InputStream inputStream = null;
        String json = “”;

        //Perform HTTP request
        try {
                HttpClient client = new DefaultHttpClient();
                HttpPost post = new HttpPost(APIlink);
                HttpResponse response = client.execute(post);
                HttpEntity entity = response.getEntity();
                inputStream = entity.getContent();
        } catch(Exception e) {}

        //Convert inputStream to String
        try {
                BufferedReader reader = new BufferedReader(new InputStreamReader(inputStream,”utf-8”),8);
                StringBuilder sbuild = new StringBuilder();
                String line = null;
                while ((line = reader.readLine()) != null){sbuild.append(line);}
                inputStream.close();
                json = sbuild.toString();
        } catch(Exception e) {}

        //Convert String to JSON object
        JSONParser parser = new JSONParser();
        Object obj = parser.parse(json);
        JSONObject jsonObj = (JSONObject) obj;

        return jsonObj;
}
```

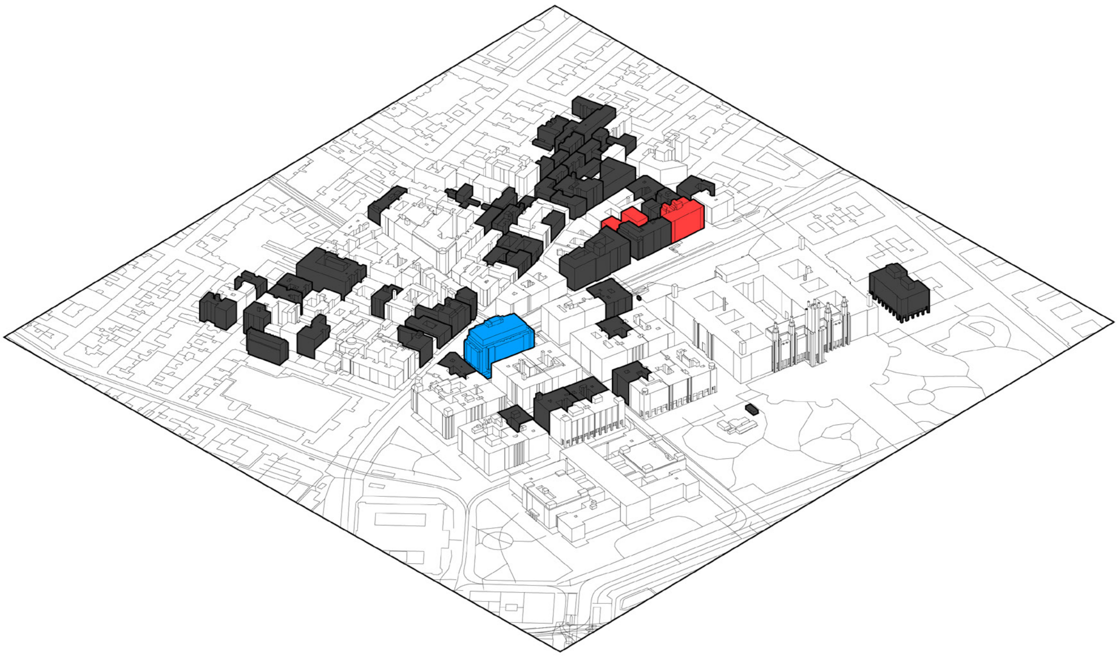
```
//Geocoding, other HTTP dependent functions work similar but with different API link and JSON parse route


public double[] geoCoding (String address){


        //Clean address String

        String cleanedAddress = address.replaceAll(“,”,””); //remove commas

        String[] splitAdress = cleanedAddress.split(“ “);

        address = splitAdress[0];

        if(splitAdress.length > 1){

                for (int i = 1; i < splitAdress.length; i++) {address = address + “+” + splitAdress[i];}

        }


        //Build request link

        String APIkey = “AIzaSyAWZZs6ac51GVpsxDzEGUO7tq_W-0qfflM”; //Hard-coded API request key

        String APIlink = “https://maps.googleapis.com/maps/api/geocode/json?address=” + address + “&key=” + APIkey;


        //Perform HTTP request

        JSONObject jsonObj = JSONObject httpRequest(APIlink);


        //Parse JSON file

        JSONArray jsonObject1 = (JSONArray) jsonObj .get(“results”);

        JSONObject jsonObject2 = (JSONObject)jsonObject1.get(0); //retrieve the most relevant result

        JSONObject jsonObject3 = (JSONObject)jsonObject2.get(“geometry”);

        JSONObject location = (JSONObject) jsonObject3.get(“location”);


        //Write information to variable

        double[] latlng = new double[2];

        latlng[0] = (double) location.get(“lat”);

        latlng[1] = (double) location.get(“lng”);


        return latlng;

}
```

B.

DIAGRAMS AND GRAPHS

Function View

Education



N

🟥 Education

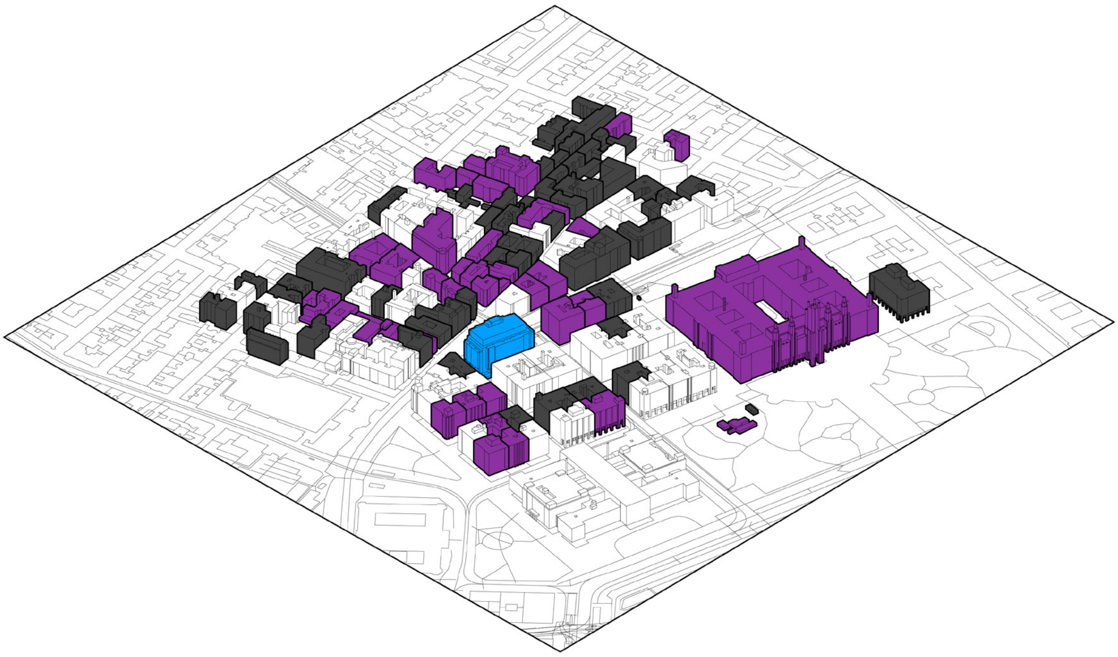
🟦 Project Site

⬛ No Data

Data availability rate of 55%

Function View

Hospitality



N

🟪 Hospitality

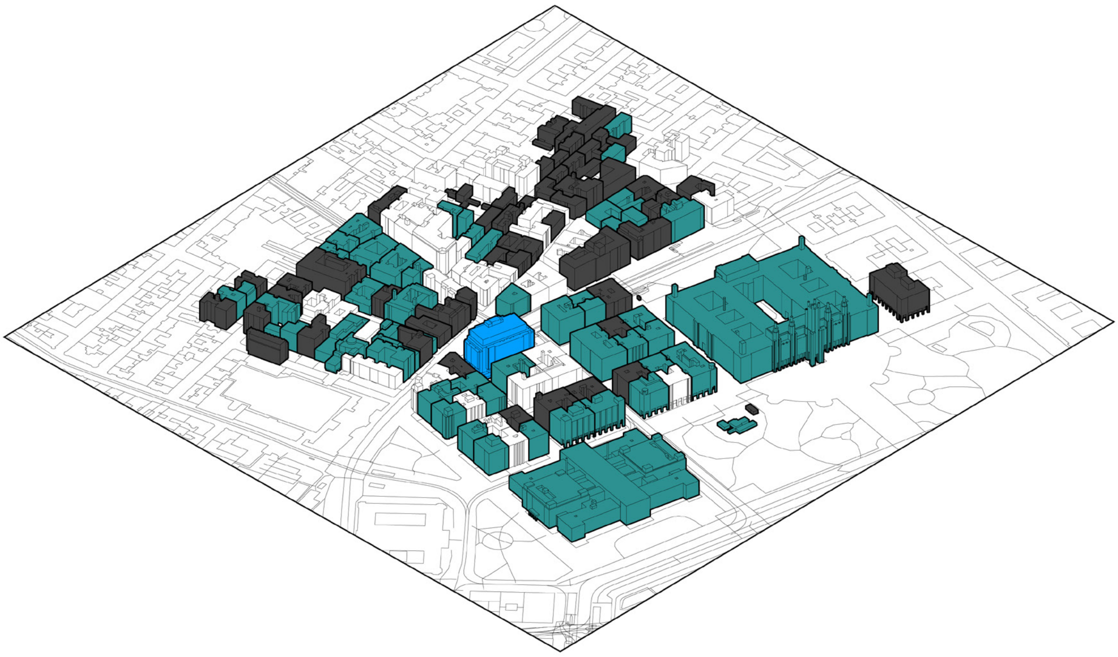
🟦 Project Site

⬛ No Data

Data availability rate of 55%

Function View

Office



N

🟩 Office

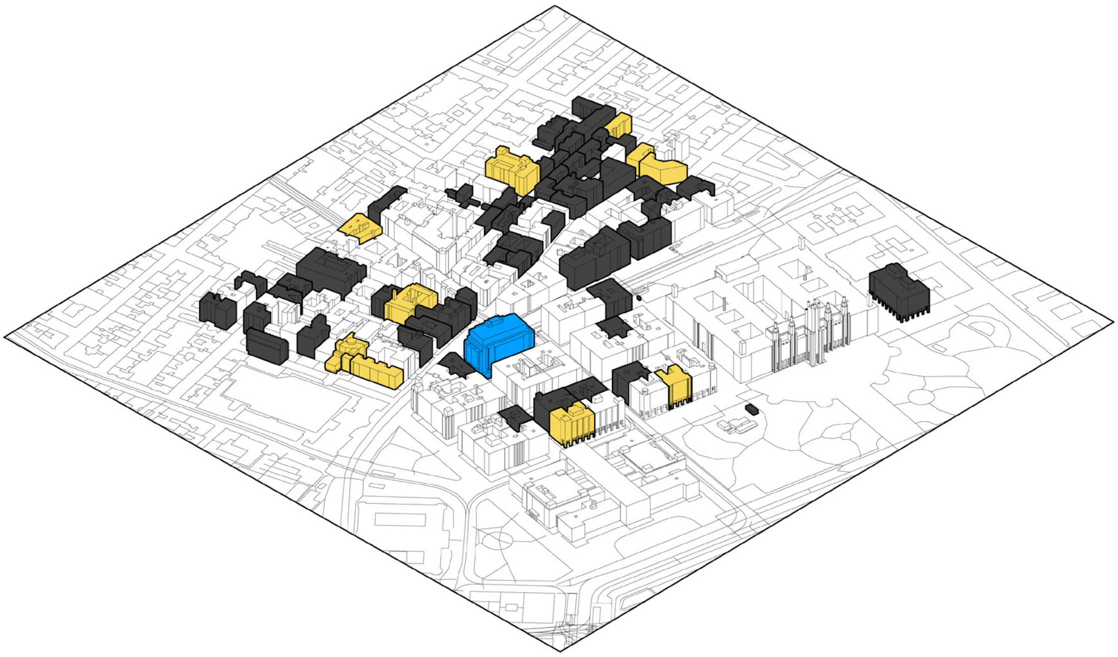
🟦 Project Site

⬛ No Data

Data availability rate of 55%

Function View

Commercial



N

🟨 Commercial

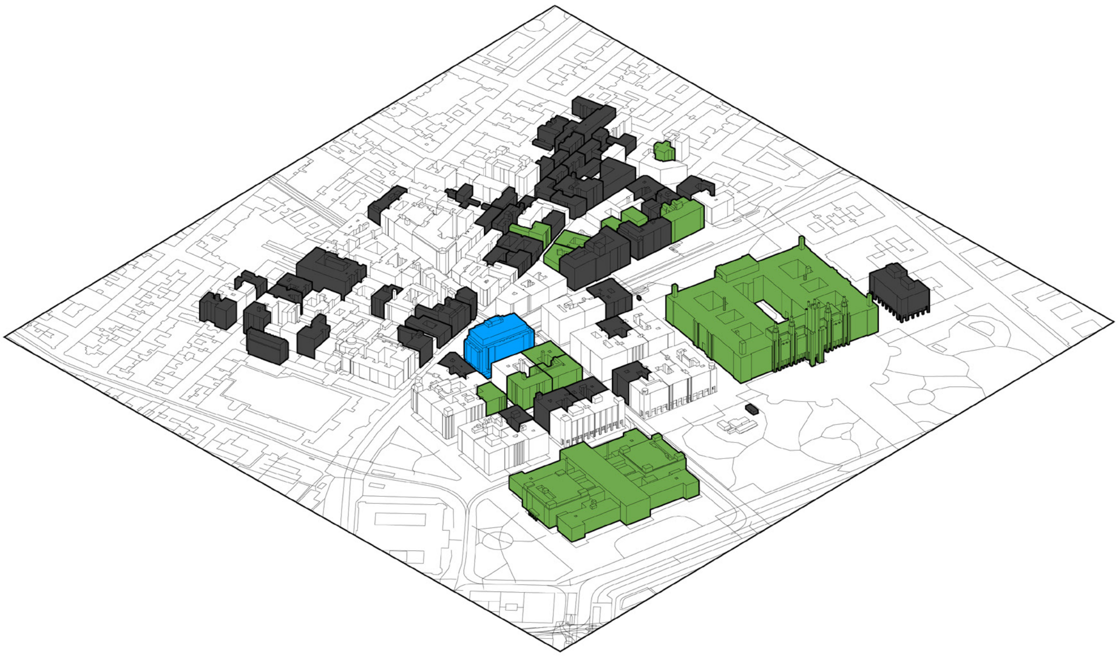
🟦 Project Site

⬛ No Data

Data availability rate of 55%

Function View

Public



N

🟩 Public

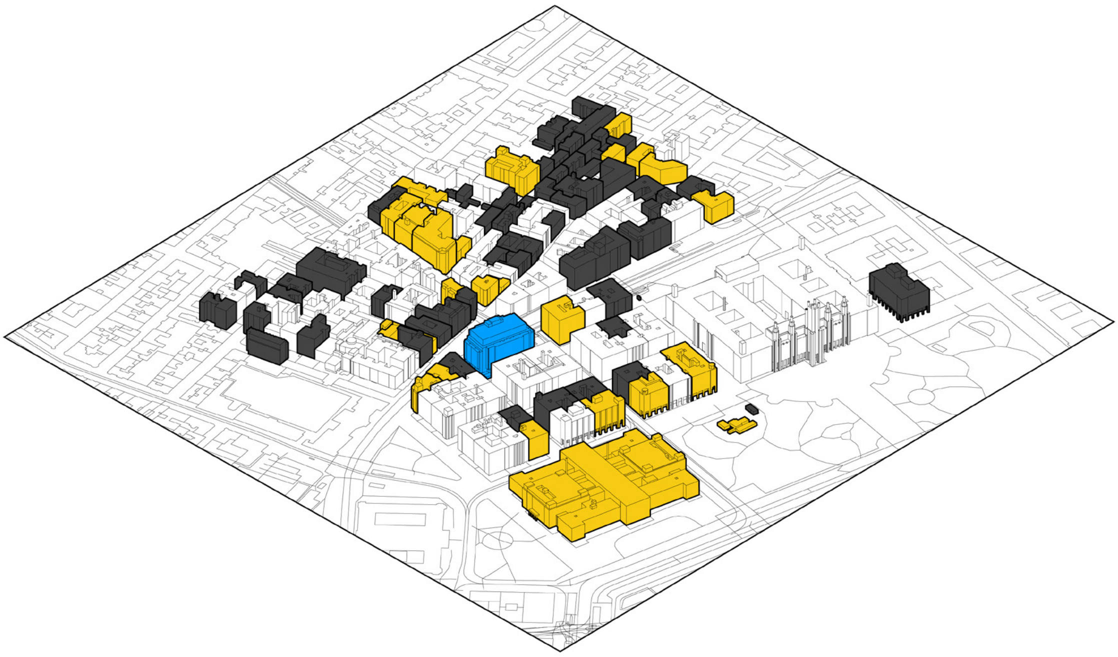
🟦 Project Site

⬛ No Data

Data availability rate of 55%

Function View

Health



N

🟨 Health

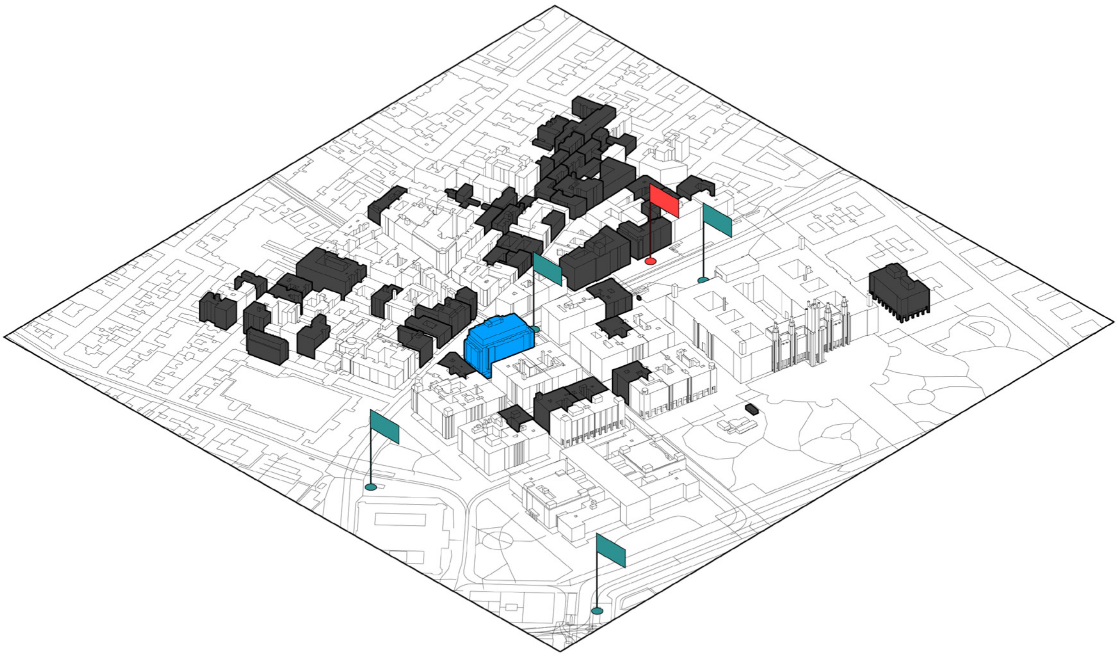
🟦 Project Site

⬛ No Data

Data availability rate of 55%

Transportation

Function View



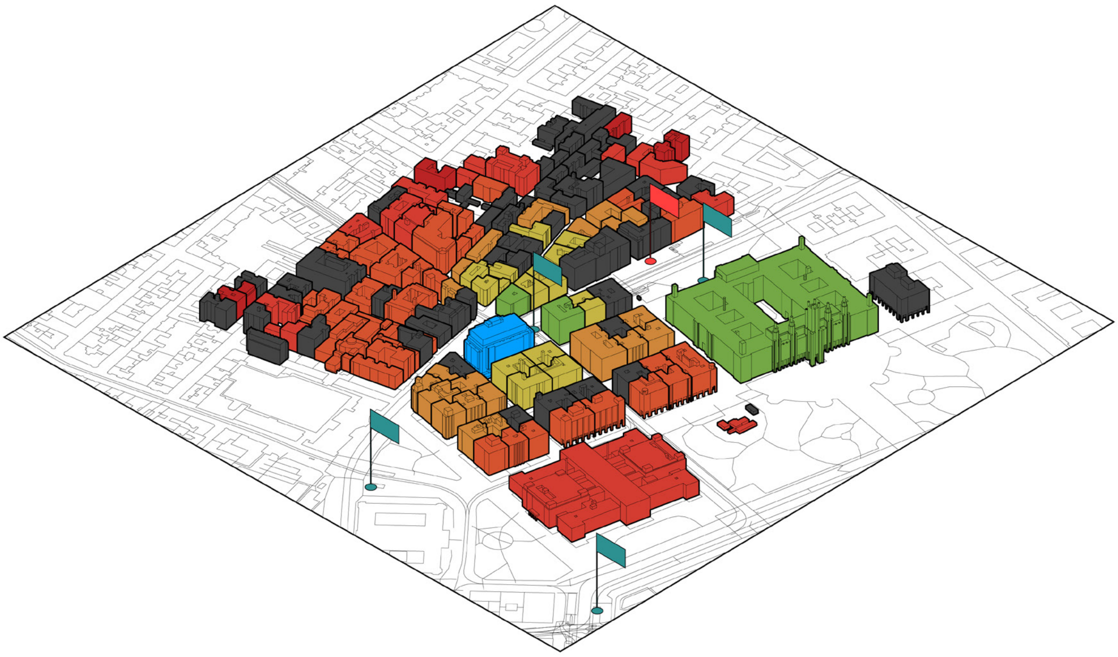
N

🟥	Metro Station	🟦	Project Site
🟩	Transit Station	⬛	No Data

Data availability rate of 55%

Transportation

Distance View



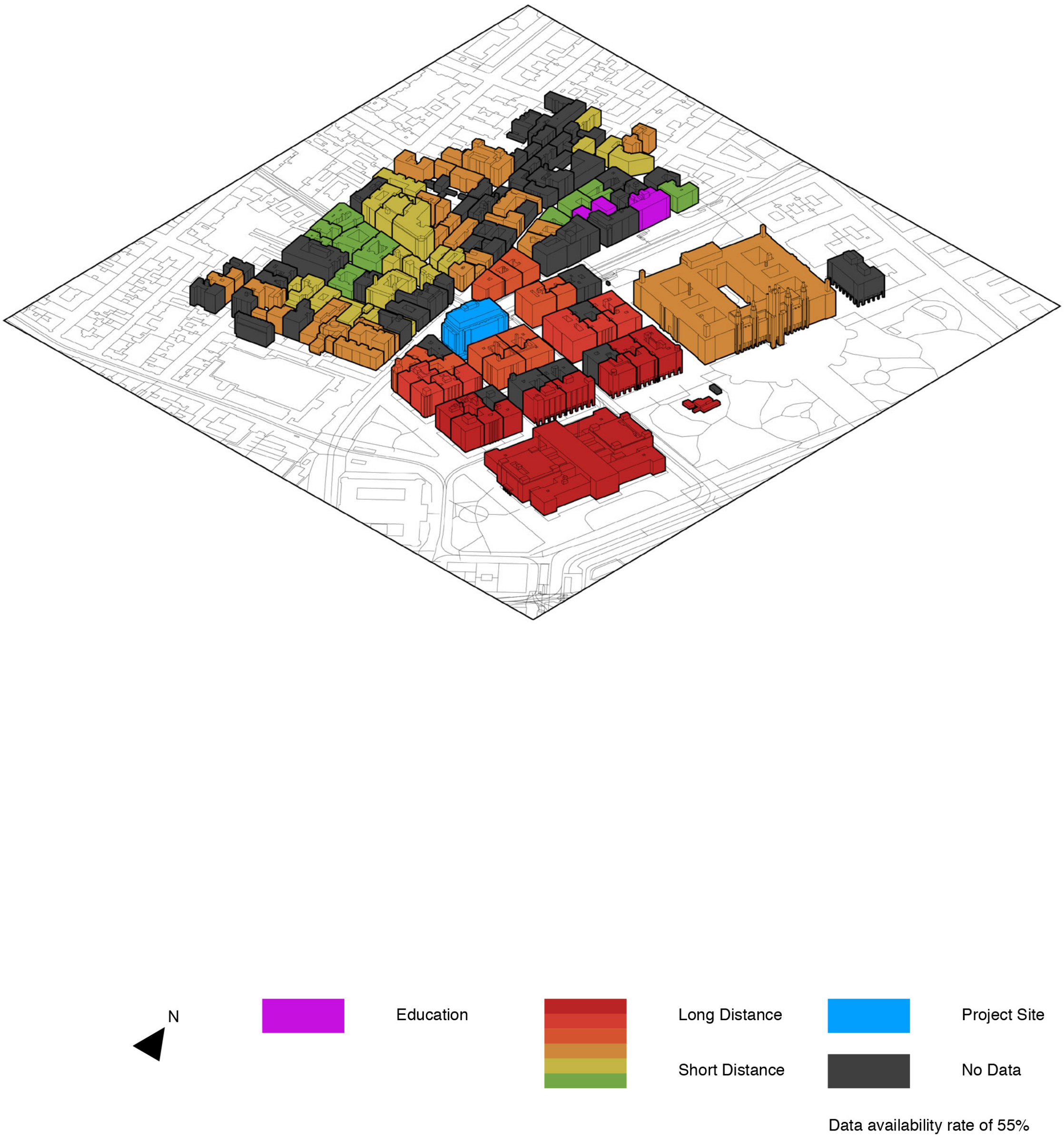
N

🟥	Metro Station	🟥	Long Distance	🟦	Project Site
🟦	Transit Station	🟩	Short Distance	⬛	No Data

Data availability rate of 55%

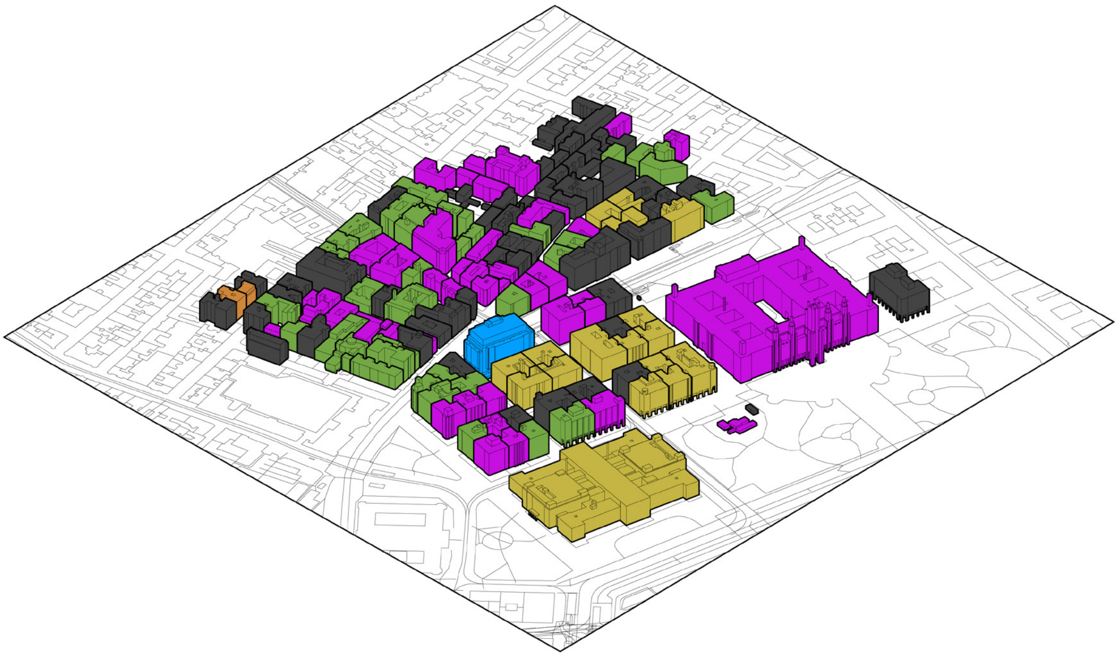
Education

Distance View



Hospitality

Distance View



N

🟪 Hospitality

🟥 Long Distance

🟩 Short Distance

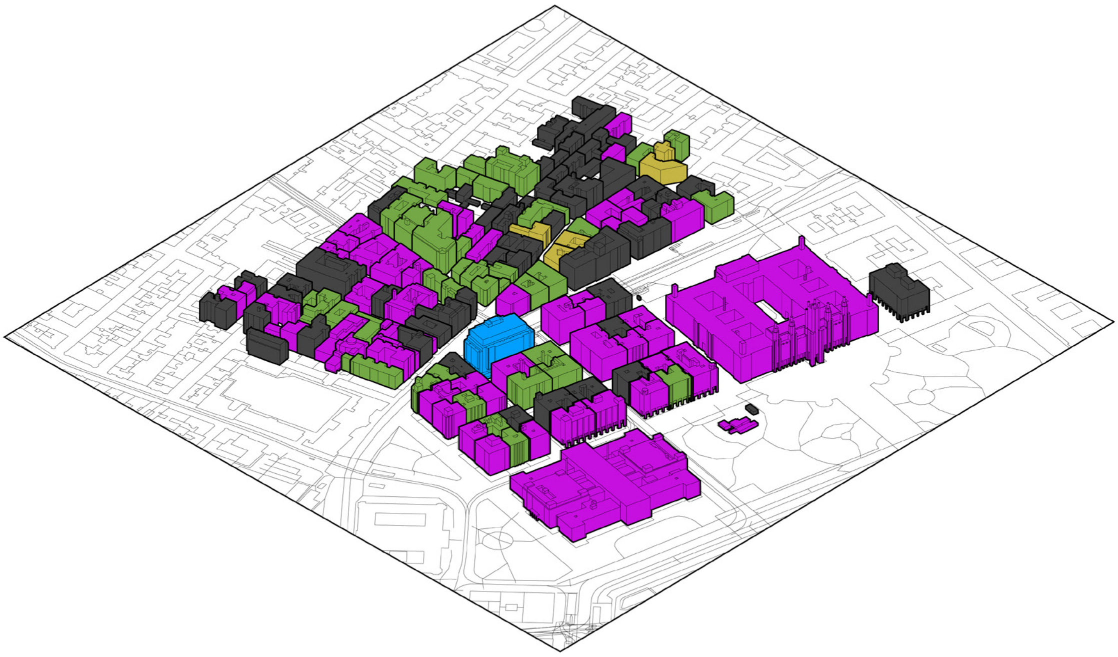
🟦 Project Site

⬛ No Data

Data availability rate of 55%

Office

Distance View



N

🟪 Office

🟥 Long Distance

🟩 Short Distance

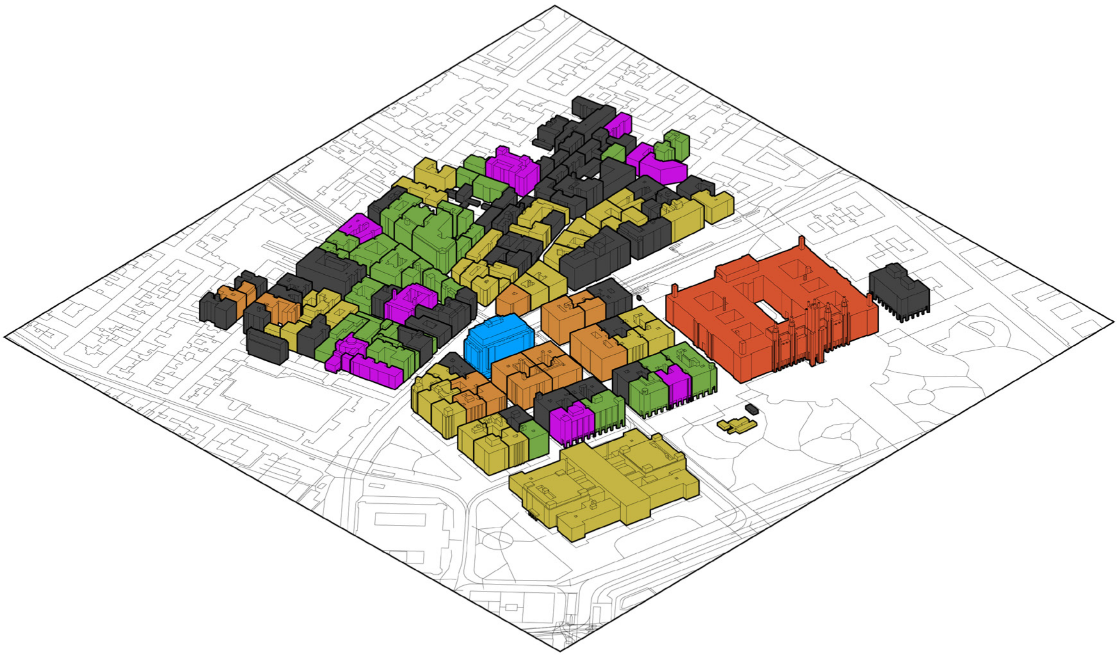
🟦 Project Site

⬛ No Data

Data availability rate of 55%

Hospitality

Distance View



N

🟪 Hospitality

🟥 Long Distance

🟩 Short Distance

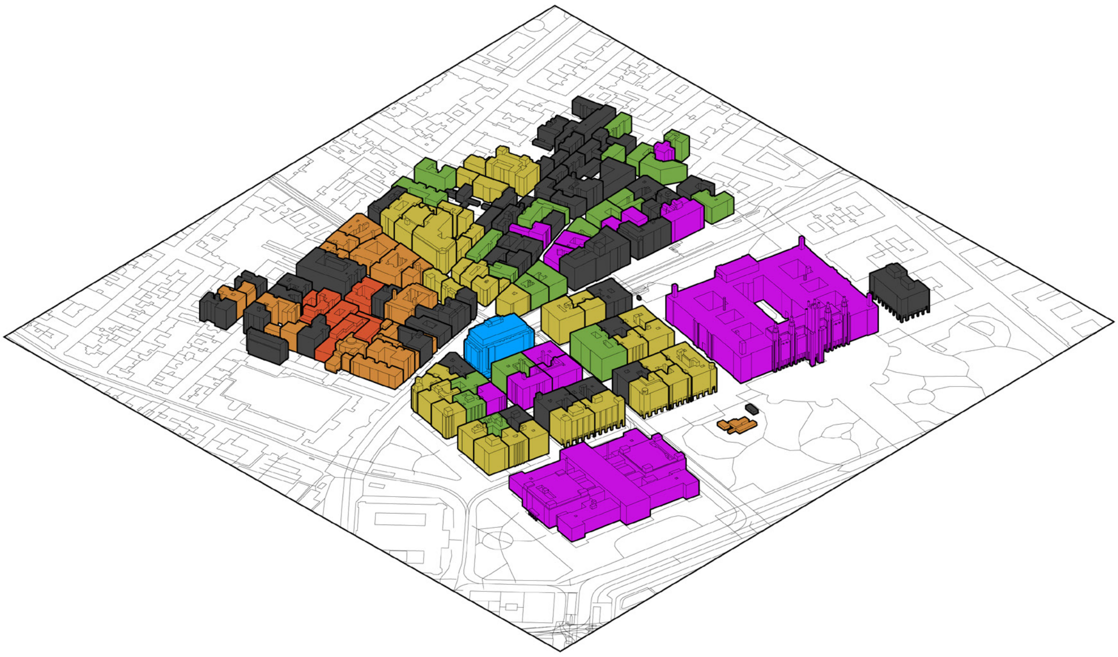
🟦 Project Site

⬛ No Data

Data availability rate of 55%

Public

Distance View



N

🟪 Public

🟥 Long Distance

🟩 Short Distance

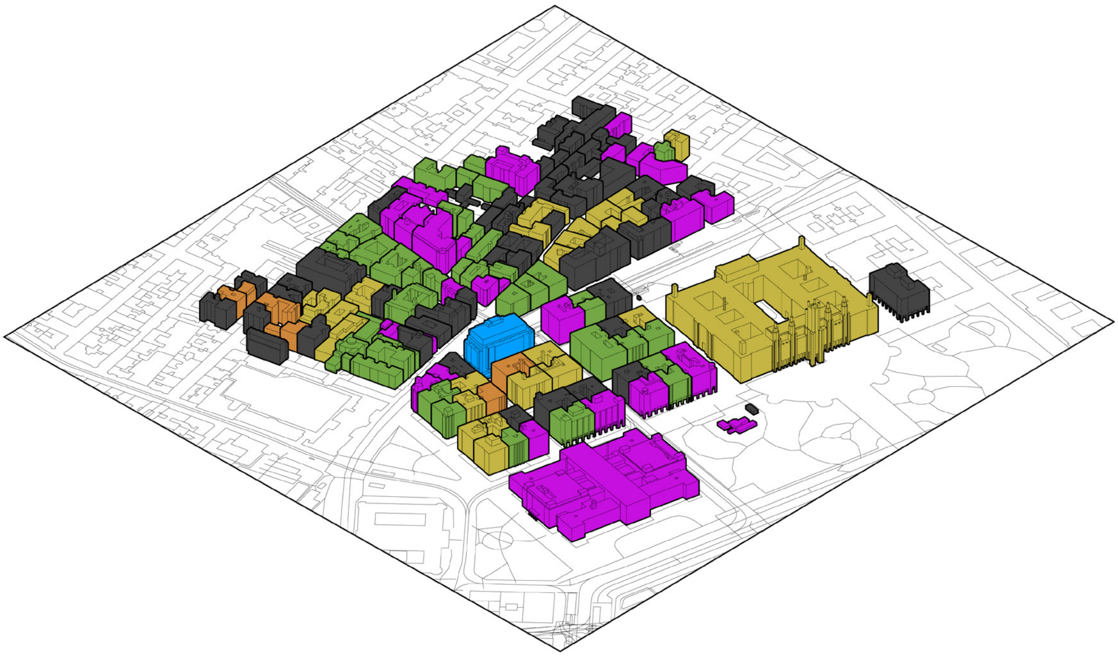
🟦 Project Site

⬛ No Data

Data availability rate of 55%

Health

Distance View

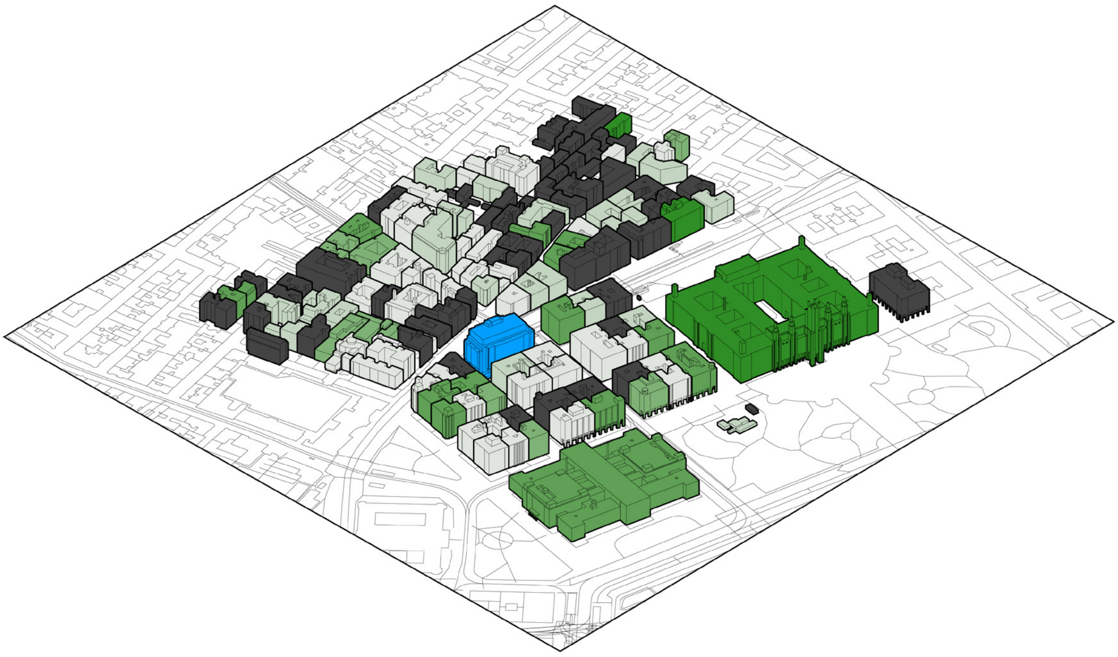


N

🟪	Health	🟥	Long Distance	🟦	Project Site
		🟩	Short Distance	⬛	No Data

Data availability rate of 55%

S-D View



N

🟩 >5 S-D objects

🟩 1 S-D object

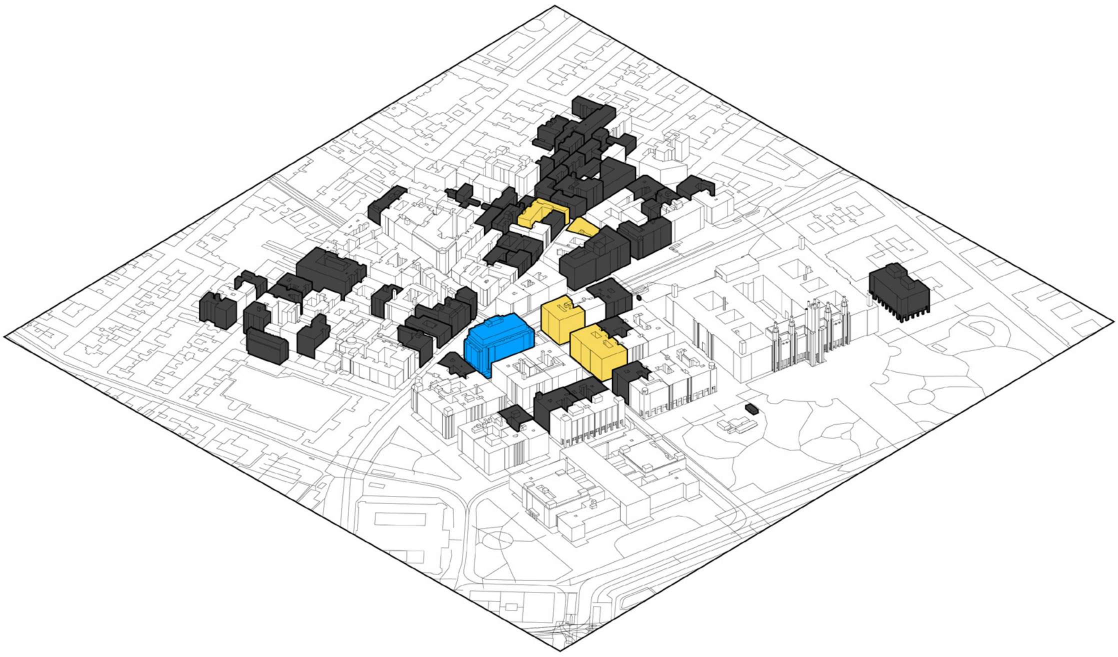
🟦 Project Site

⬛ No Data

Data availability rate of 55%

Time View

6 AM



N

🟨 Activity

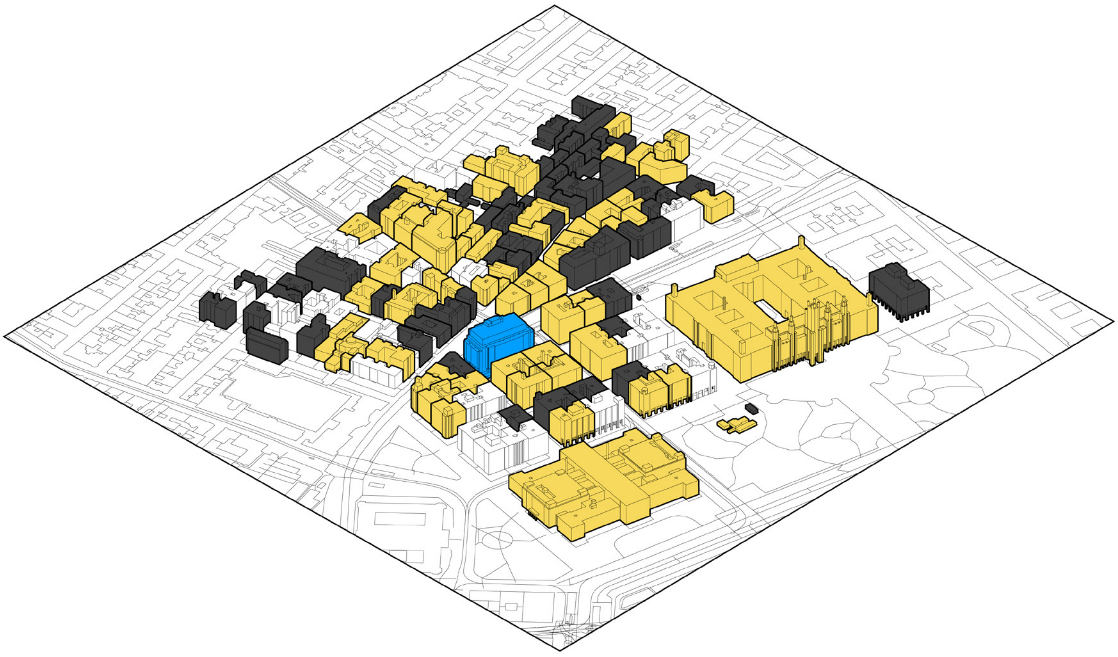
🟦 Project Site

⬛ No Data

Data availability rate of 55%

12 AM

Time View



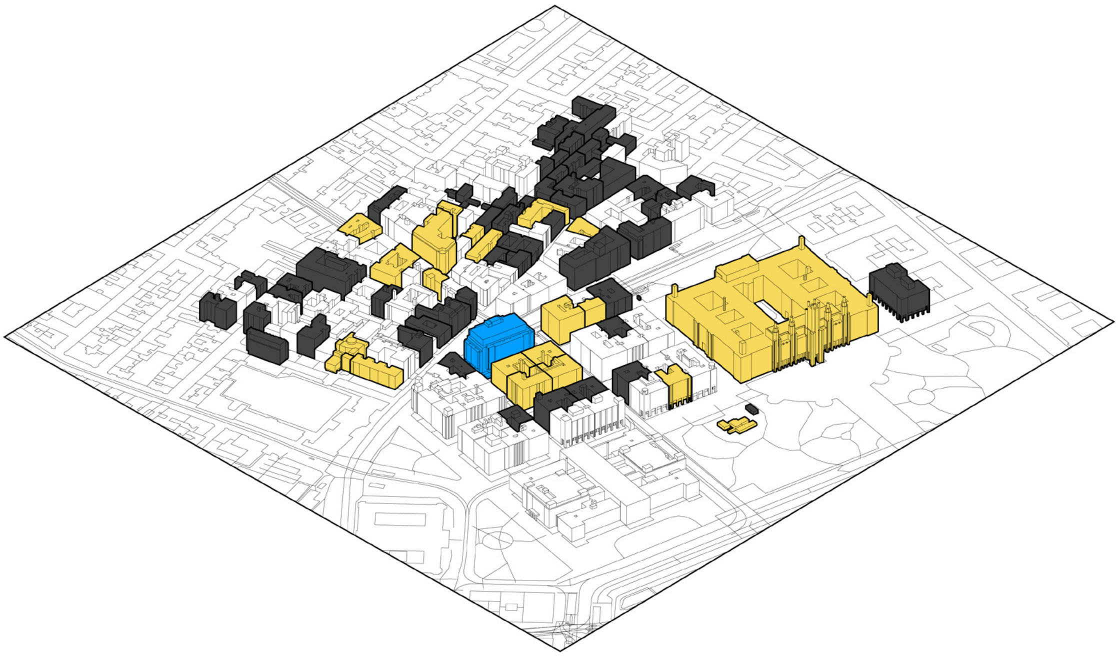
N

🟨 Activity

🟦 Project Site

⬛ No Data

Data availability rate of 55%



N

🟨 Activity

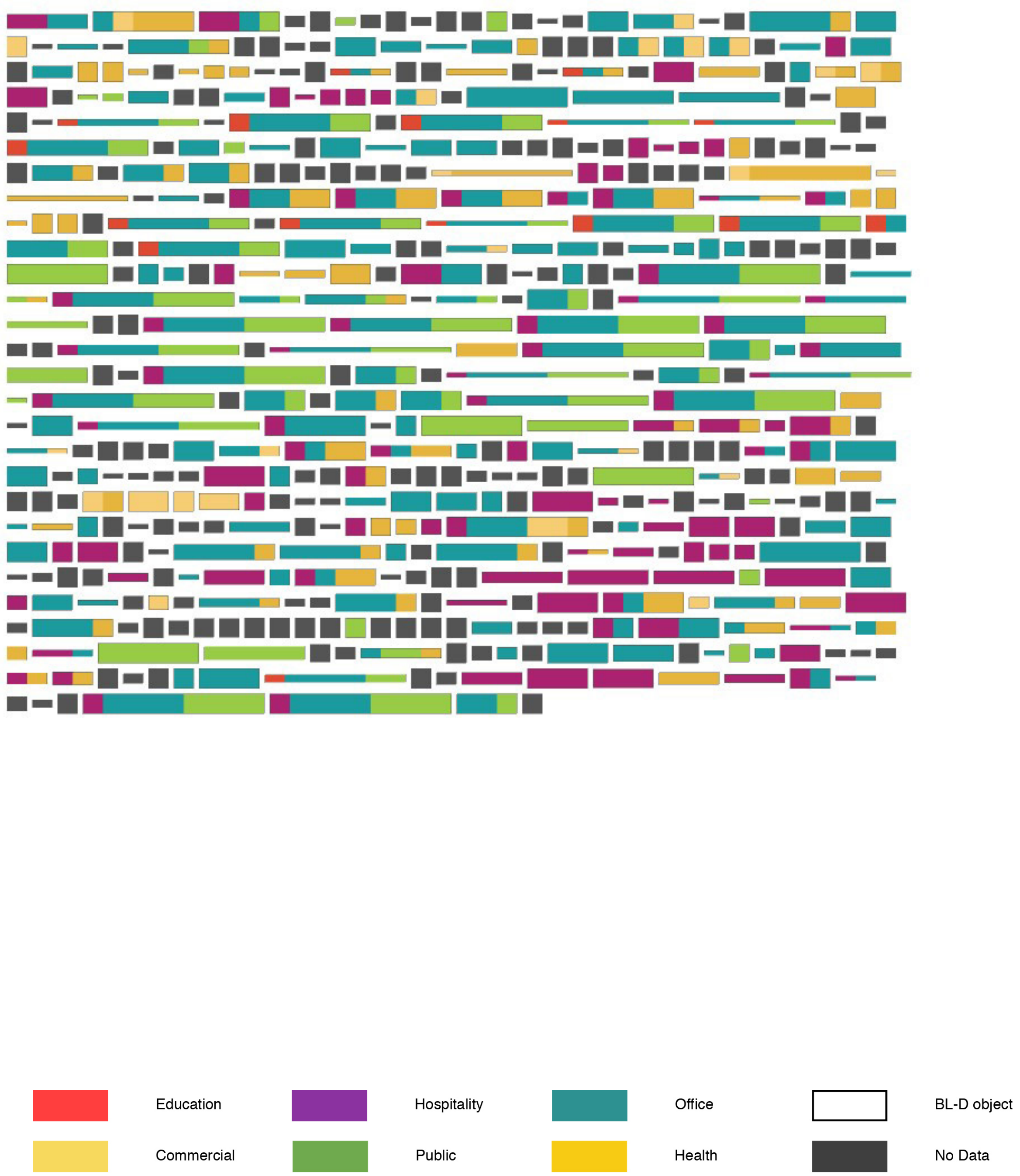
🟦 Project Site

⬛ No Data

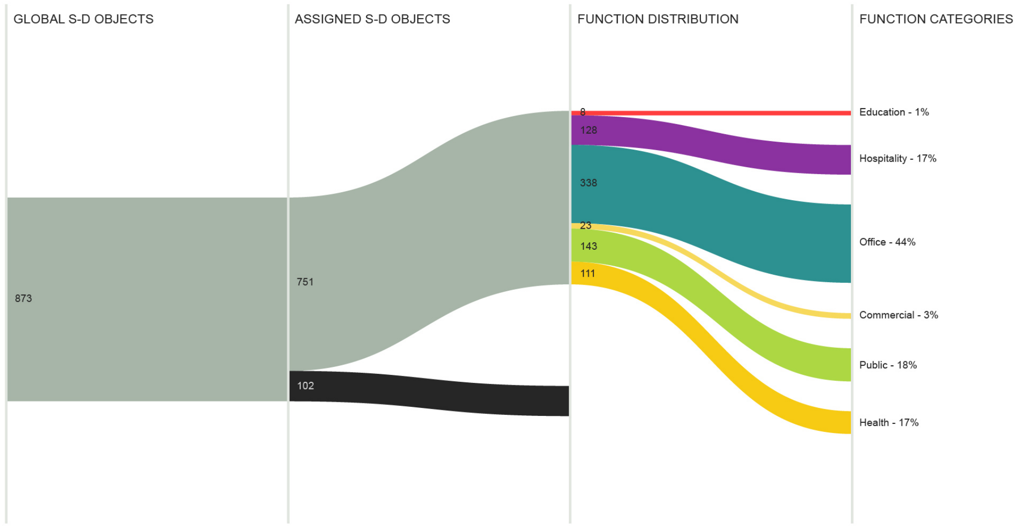
Data availability rate of 55%

Time View

6 PM

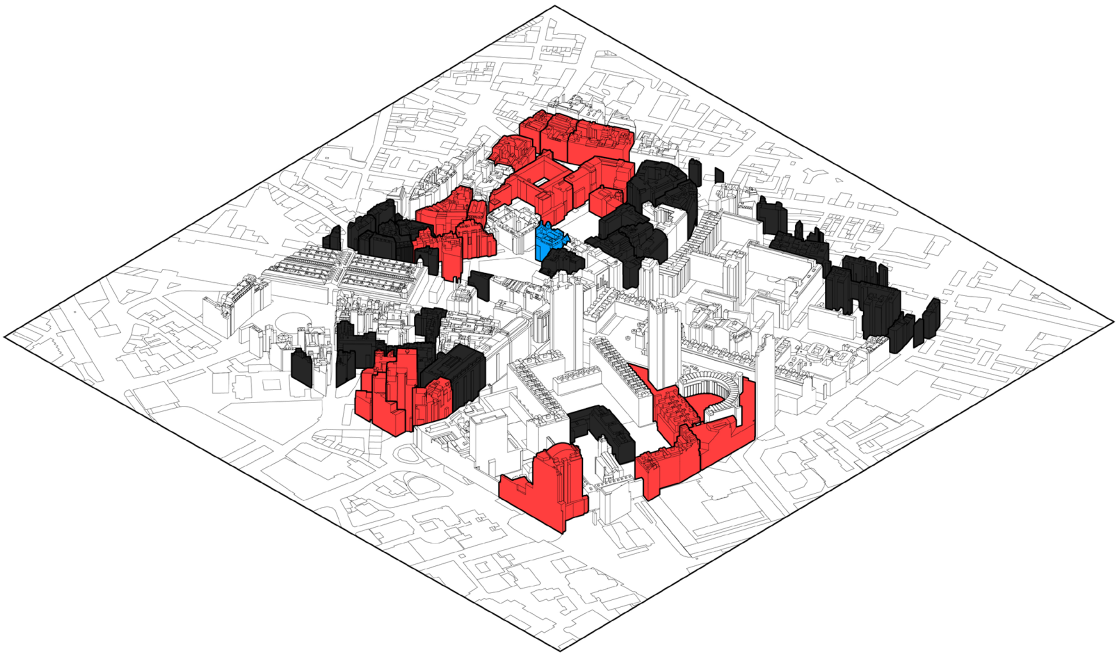


String height represents relative mesh volume of parent BL-D object


GLOBAL S-D OBJECTS
ASSIGNED S-D OBJECTS
FUNCTION DISTRIBUTION
FUNCTION CATEGORIES
873
751
102
8
128
338
23
143
111
Education - 1%
Hospitality - 17%
Office - 44%
Commercial - 3%
Public - 18%
Health - 17%

Function View

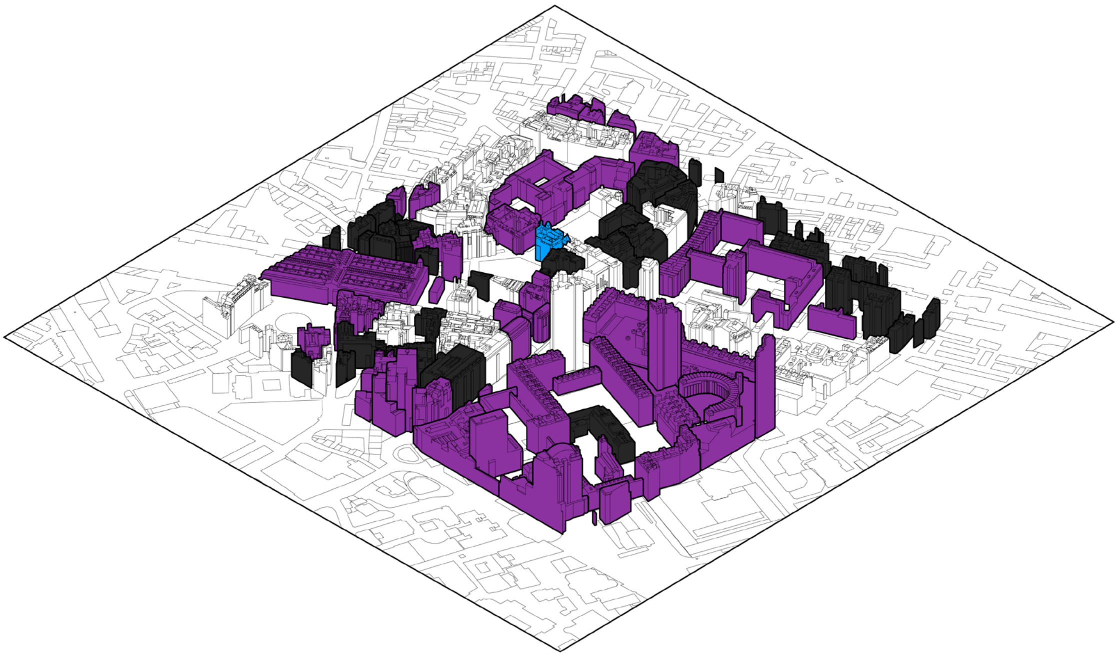
Education



N

🟥	Education	🟦	Project Site
		⬛	No Data

Data availability rate of 59%



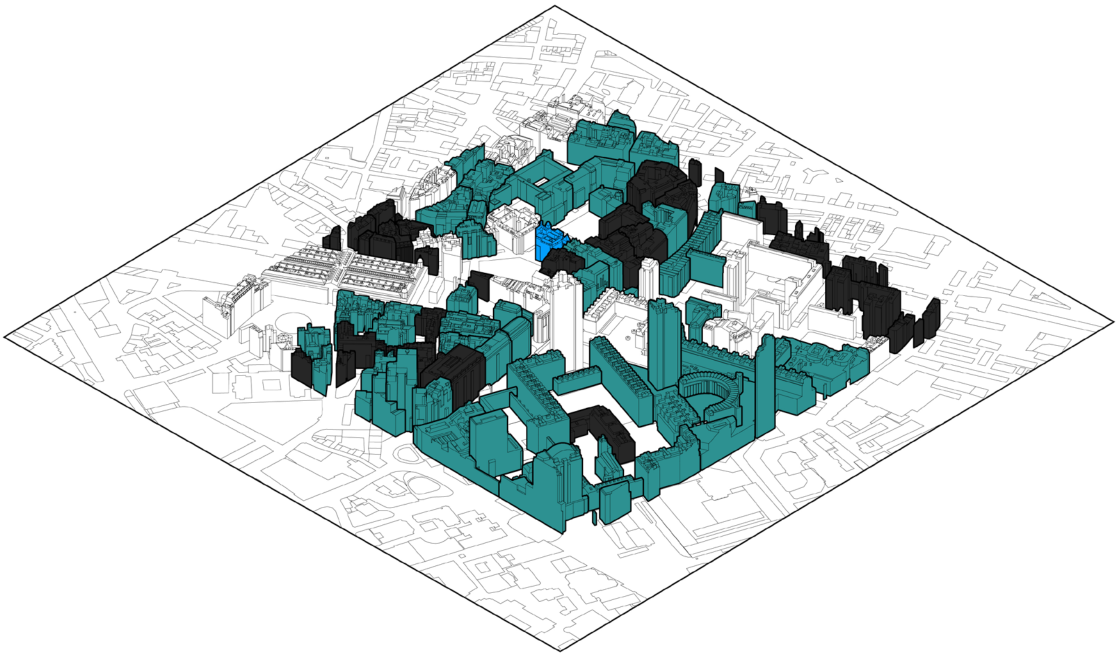
N

🟪	Hospitality	🟦	Project Site
		⬛	No Data

Data availability rate of 59%

Hospitality

Function View



N

🟩 Office

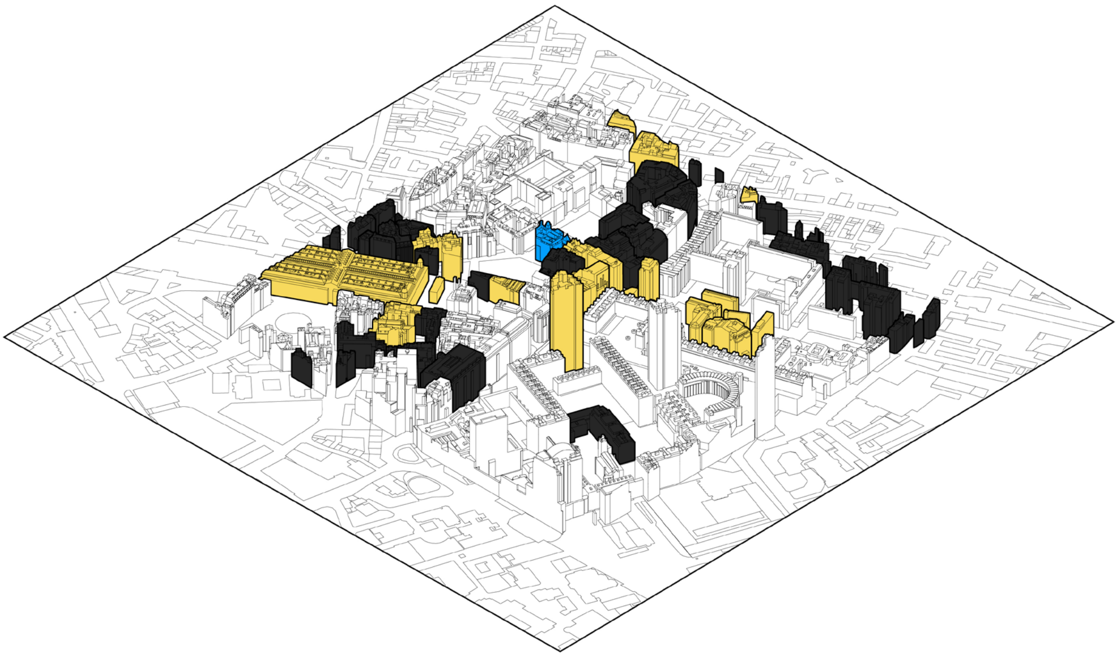
🟦 Project Site

⬛ No Data

Data availability rate of 59%

Function View

Office



Function View

Commercial

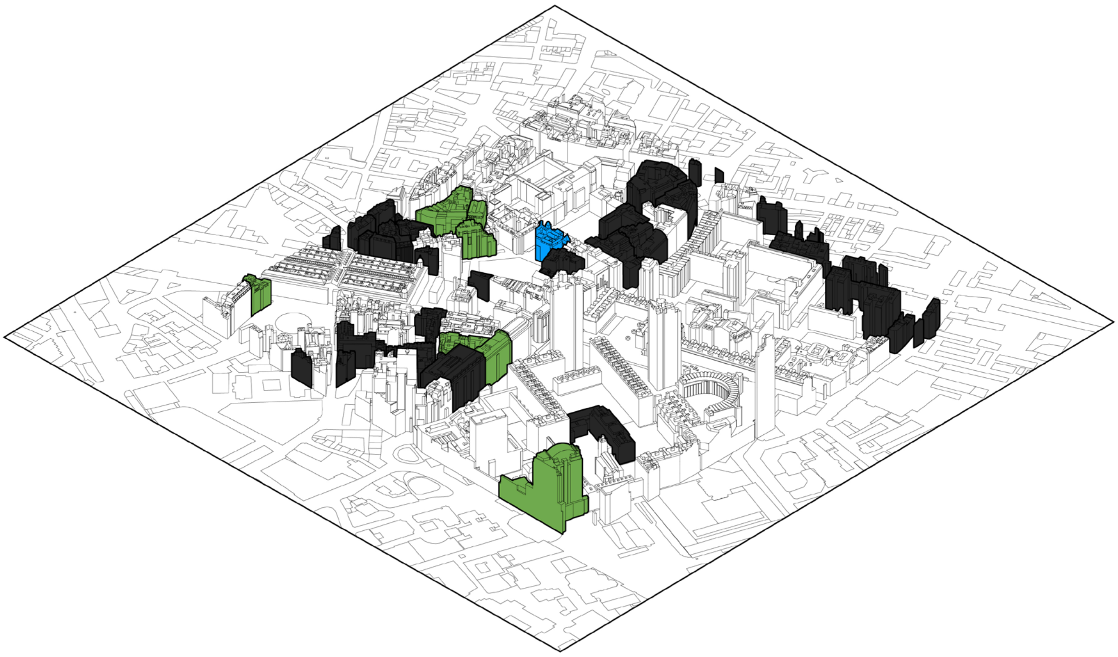
N

🟨 Commercial

🟦 Project Site

⬛ No Data

Data availability rate of 59%



N

🟩 Public

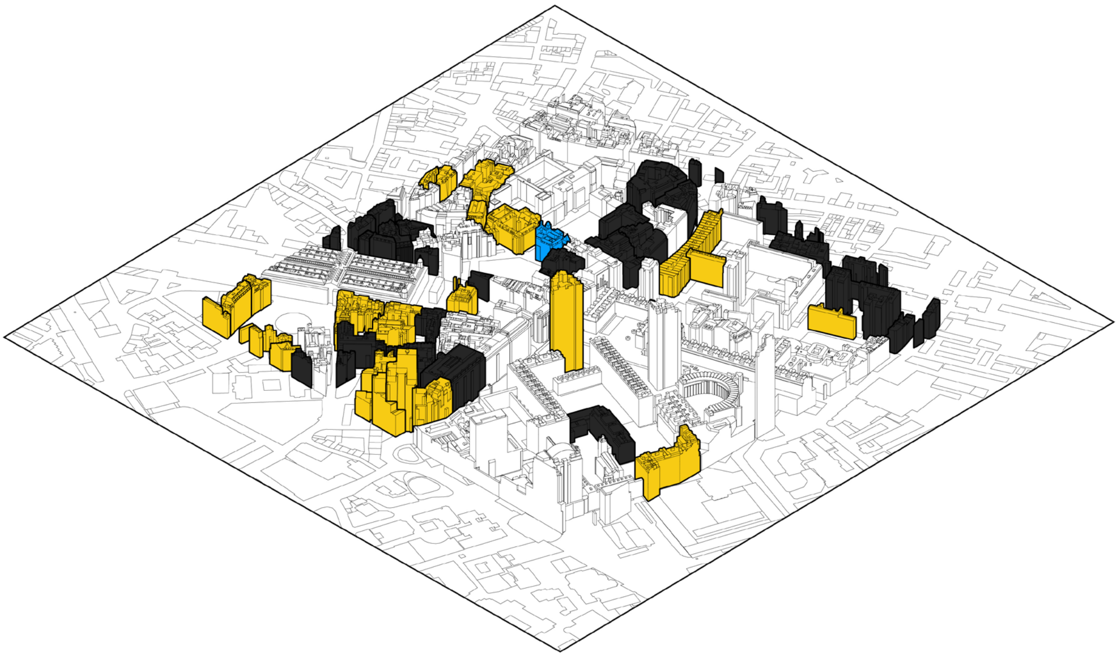
🟦 Project Site

⬛ No Data

Data availability rate of 59%

Public

Function View

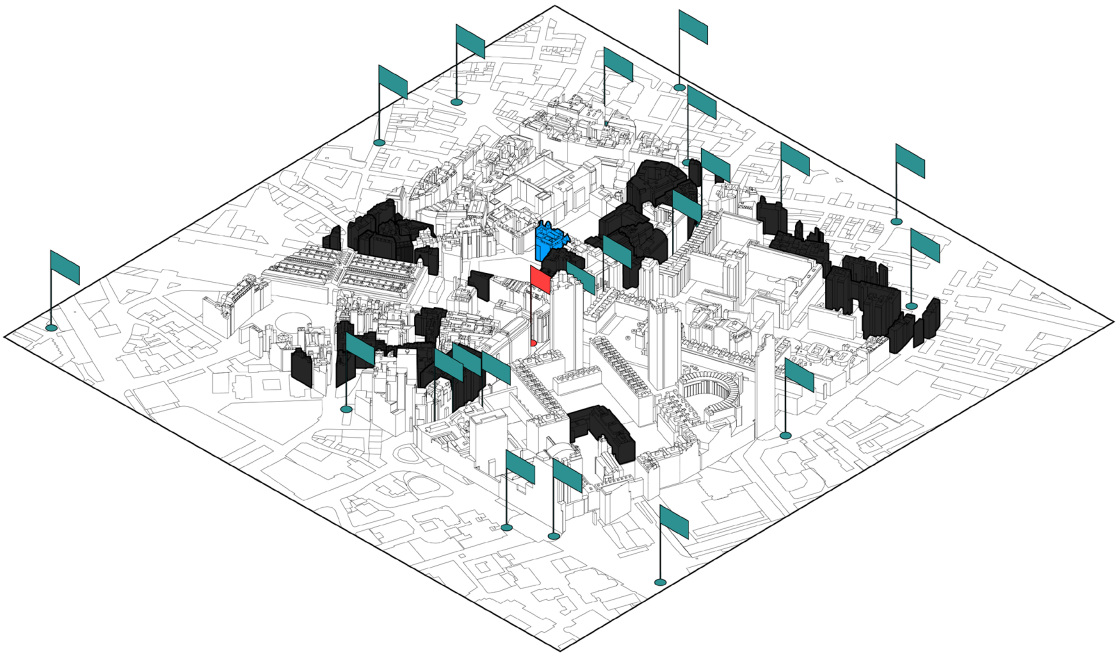


🟨	Health	🟦	Project Site
		⬛	No Data

Data availability rate of 59%

Health

Function View



🟥	Metro Station	🟦	Project Site
🟩	Transit Station	⬛	No Data

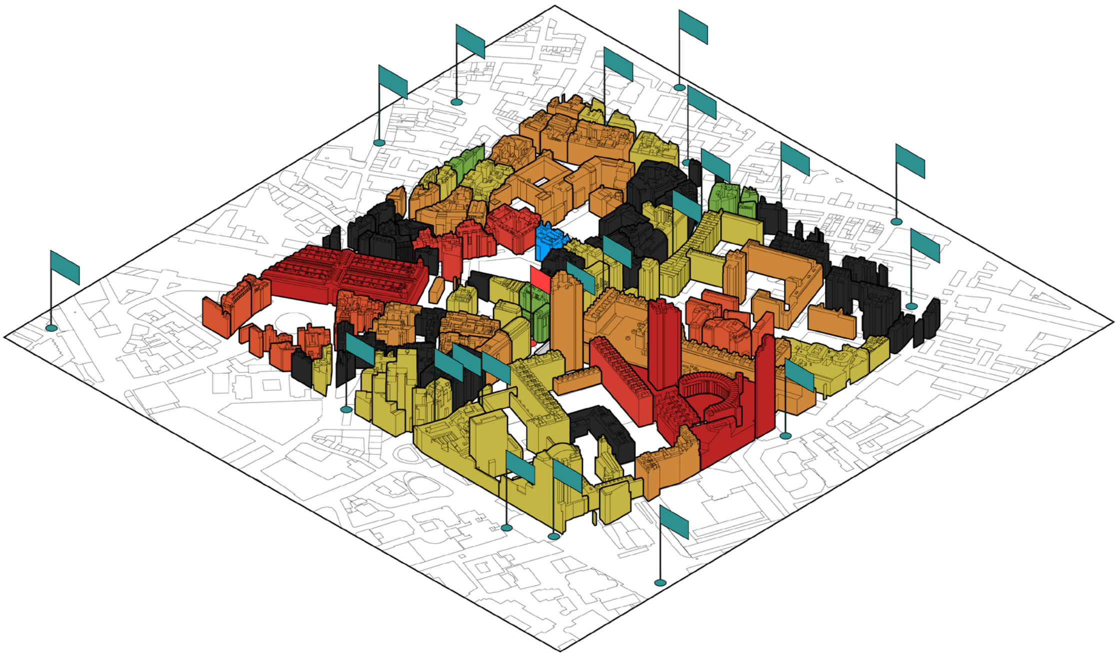
Data availability rate of 59%

Transportation

Function View

Transportation

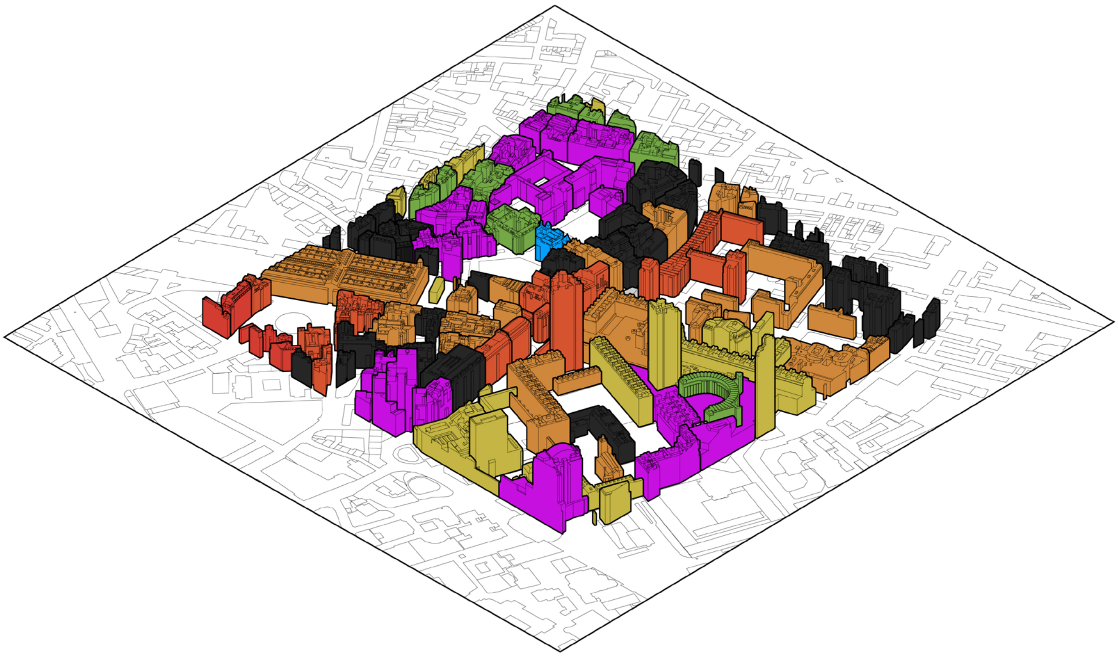
Distance View



N

🟥	Metro Station	🟥	Long Distance	🟦	Project Site
🟦	Transit Station	🟩	Short Distance	⬛	No Data

Data availability rate of 59%



N

🟪 Education

🟥 Long Distance

🟩 Short Distance

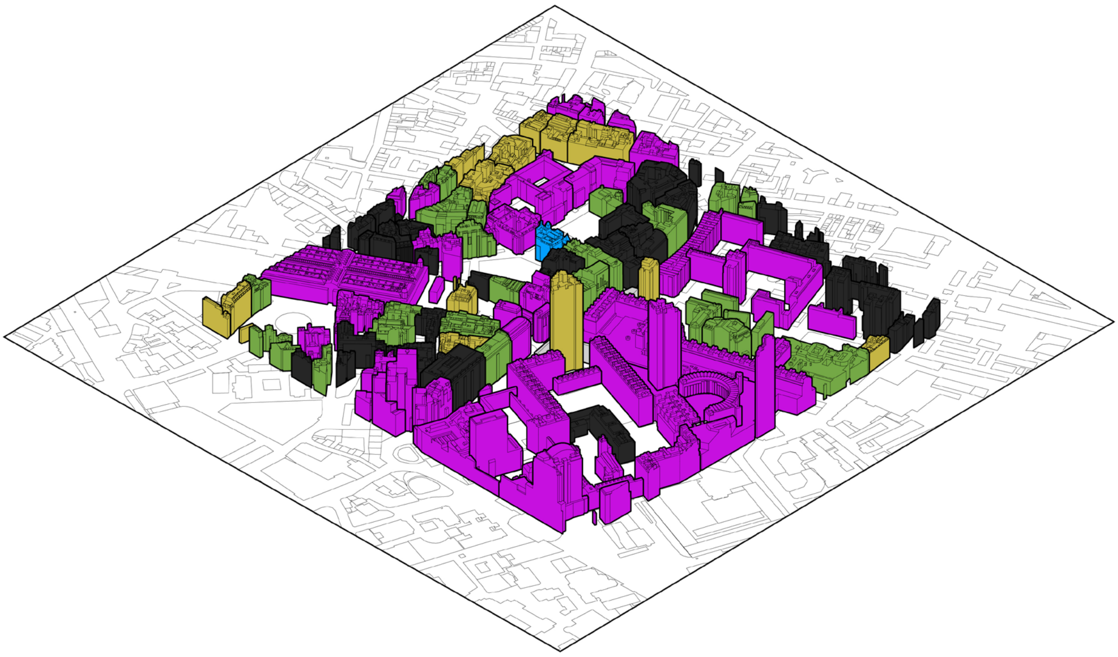
🟦 Project Site

⬛ No Data

Data availability rate of 59%

Education

Distance View



N

🟪	Hospitality	🟥	Long Distance
		🟩	Short Distance
🟦	Project Site	⬛	No Data

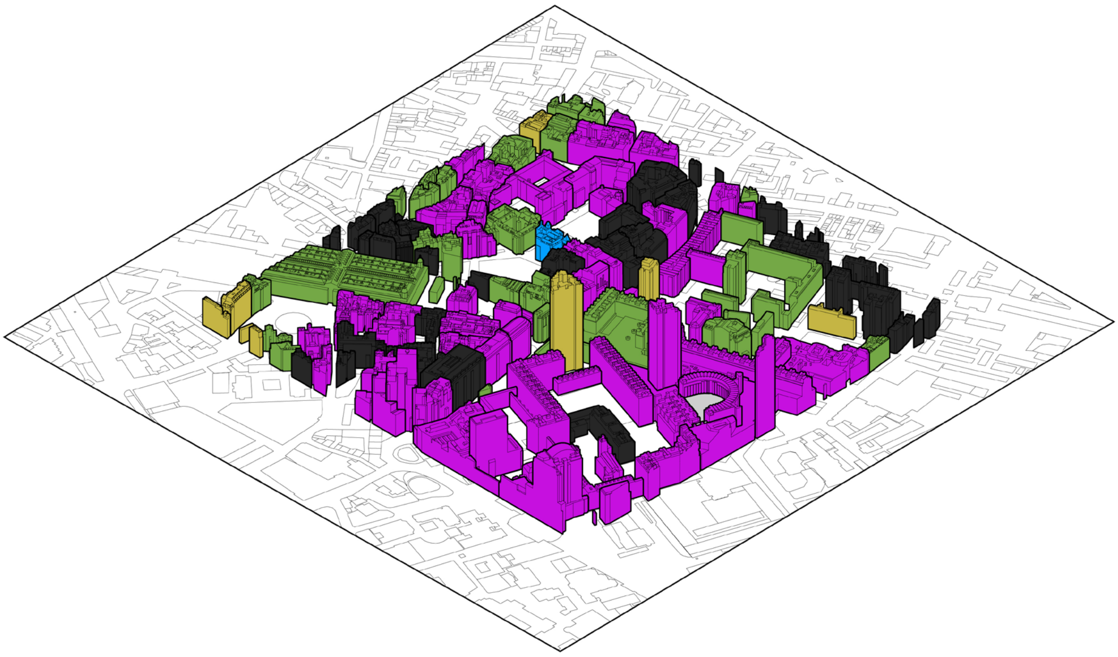
Data availability rate of 59%

Hospitality

Distance View

Office

Distance View



N

🟪 Office

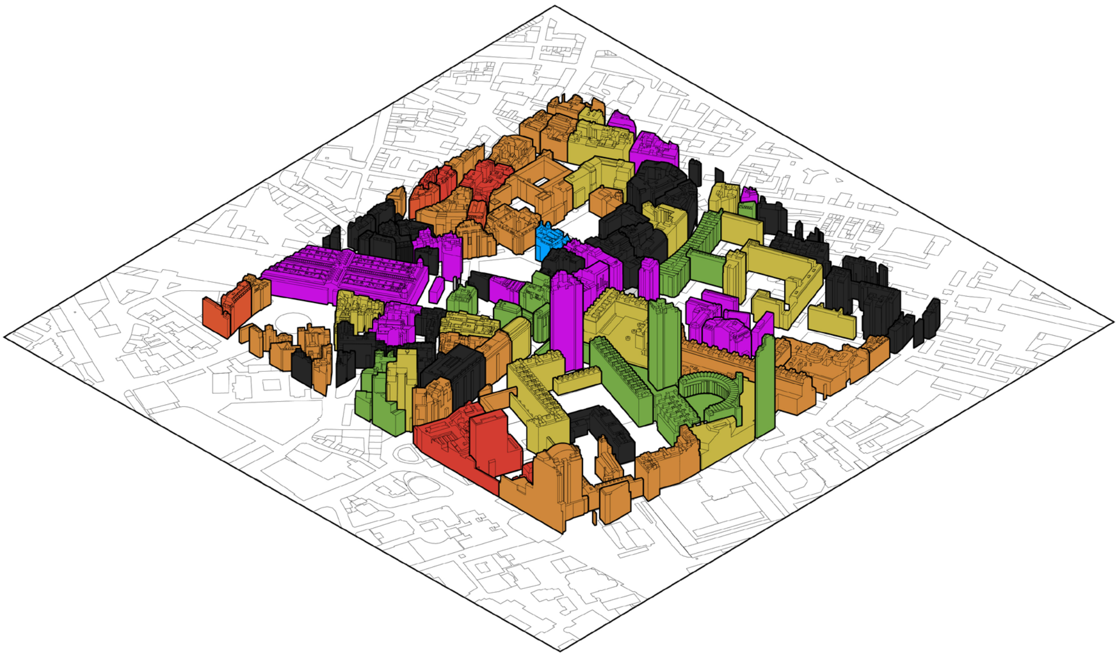
Long Distance

Short Distance

🟦 Project Site

⬛ No Data

Data availability rate of 59%



N

🟪 Commercial

🟥 Long Distance

🟩 Short Distance

🟦 Project Site

⬛ No Data

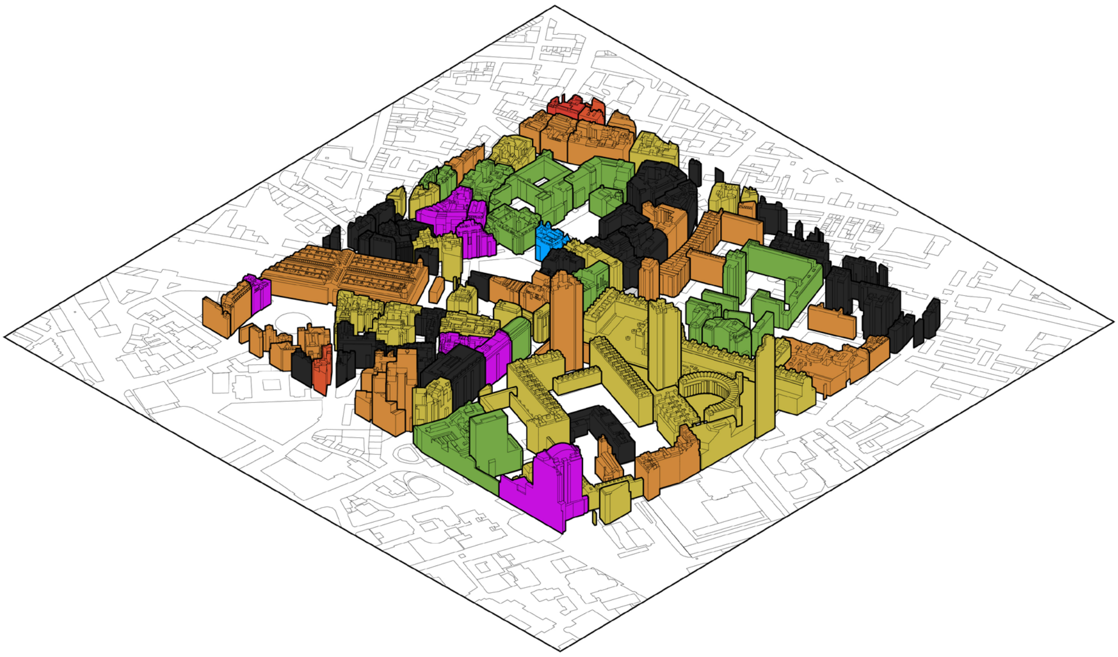
Data availability rate of 59%

Commercial

Distance View

Public

Distance View



N

🟪 Public

🟥 Long Distance

🟩 Short Distance

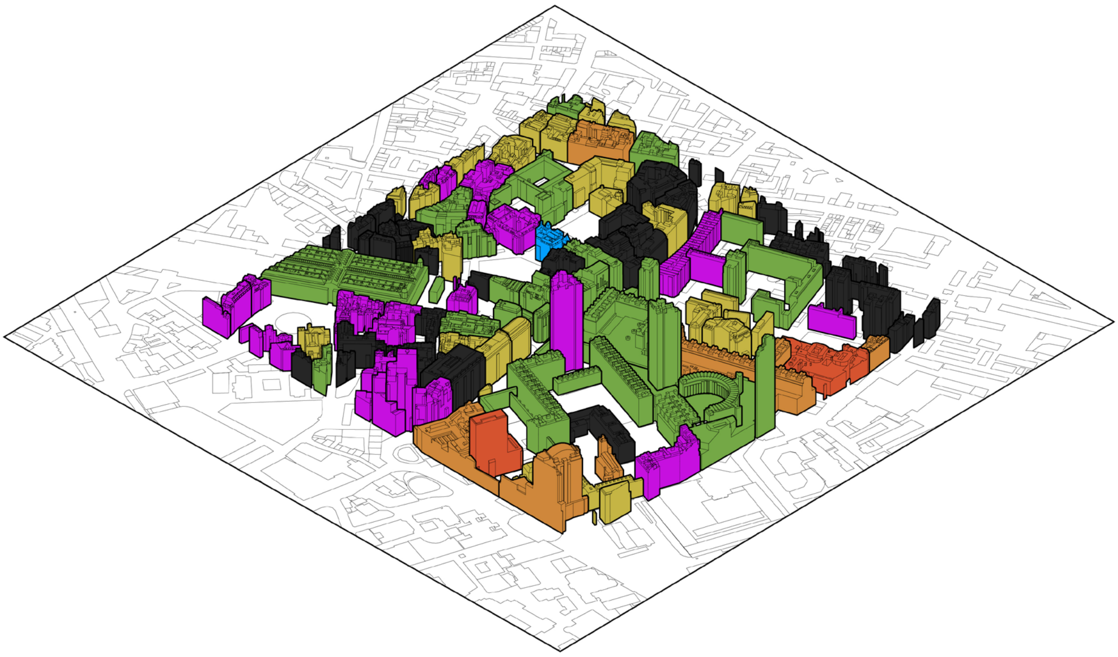
🟦 Project Site

⬛ No Data

Data availability rate of 59%

Health

Distance View



N

🟪 Health

🟥 Long Distance

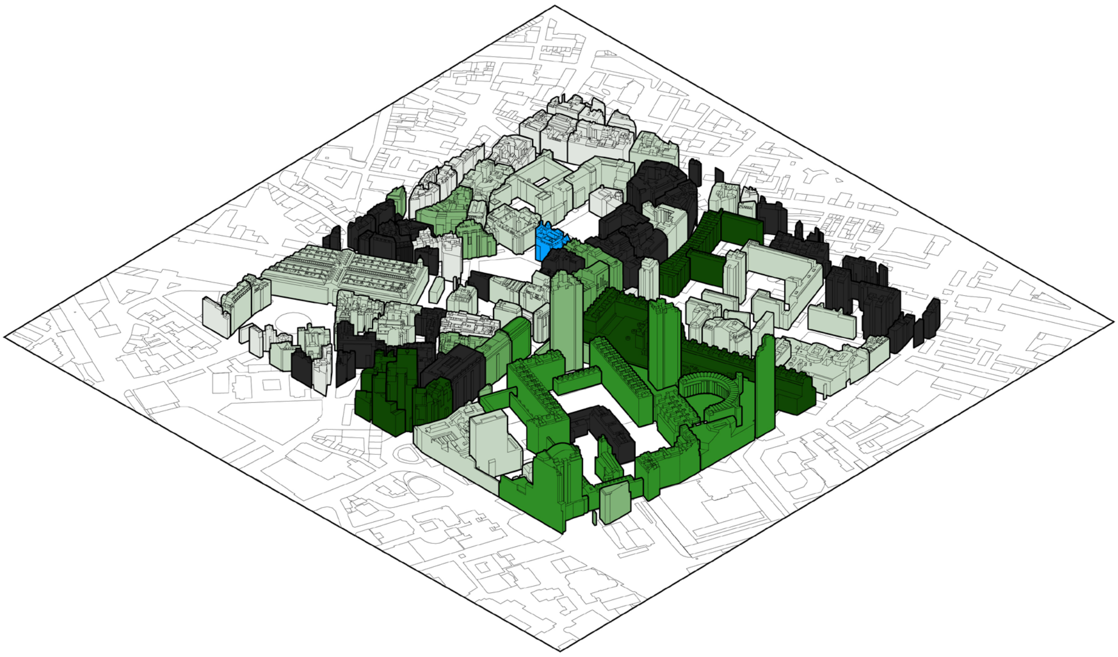
🟩 Short Distance

🟦 Project Site

⬛ No Data

Data availability rate of 59%

S-D View



N

🟩 >5 S-D objects

1 S-D object

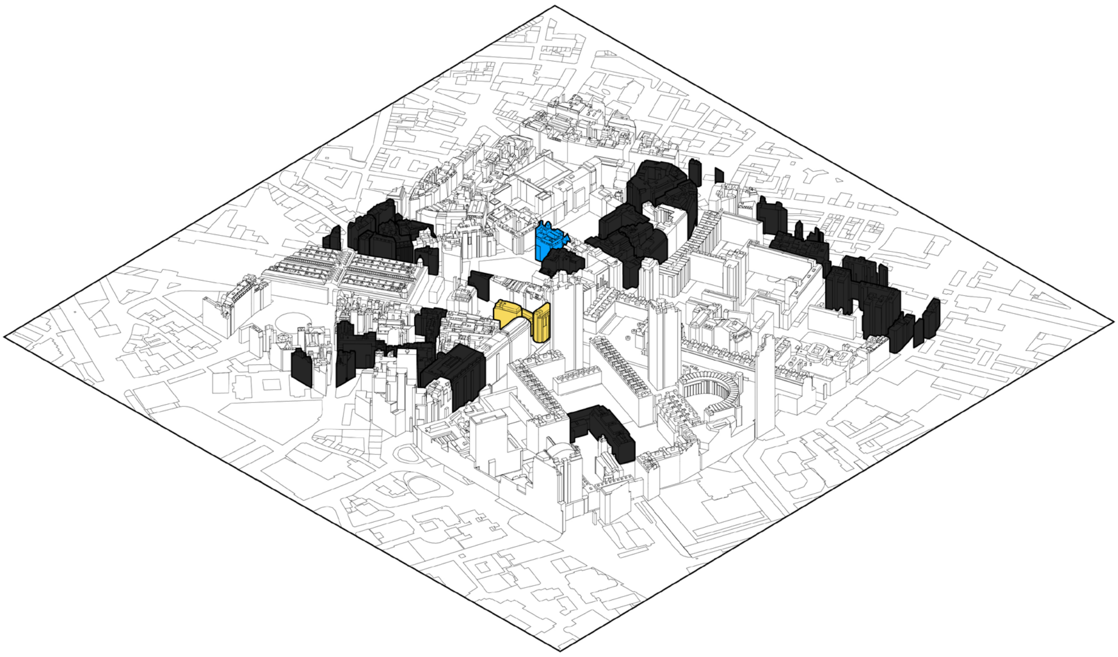
🟦 Project Site

⬛ No Data

Data availability rate of 59%

Time View

6 AM



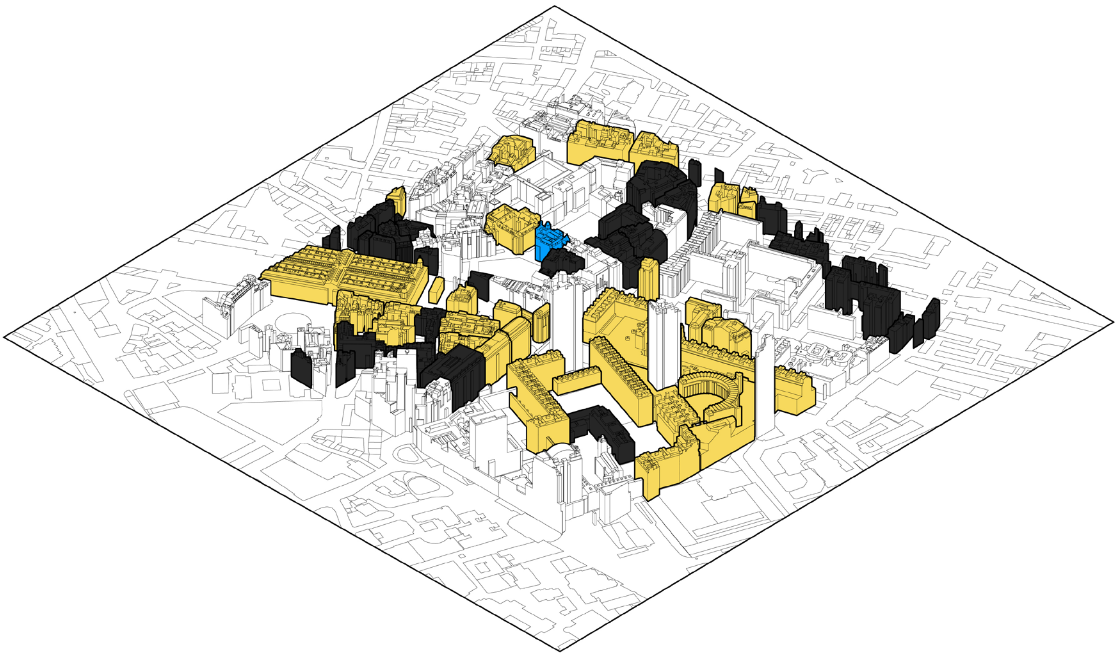
N

🟨	Activity	🟦	Project Site
		⬛	No Data

Data availability rate of 59%

Time View

12 AM



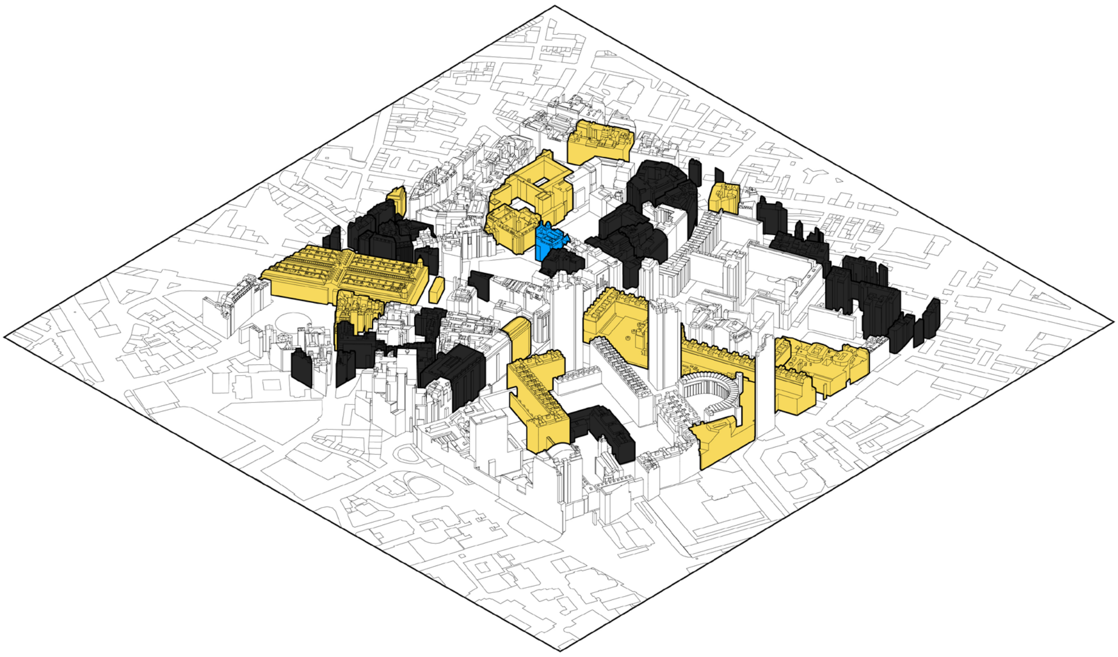
N

🟨 Activity

🟦 Project Site

⬛ No Data

Data availability rate of 59%



N

🟨 Activity

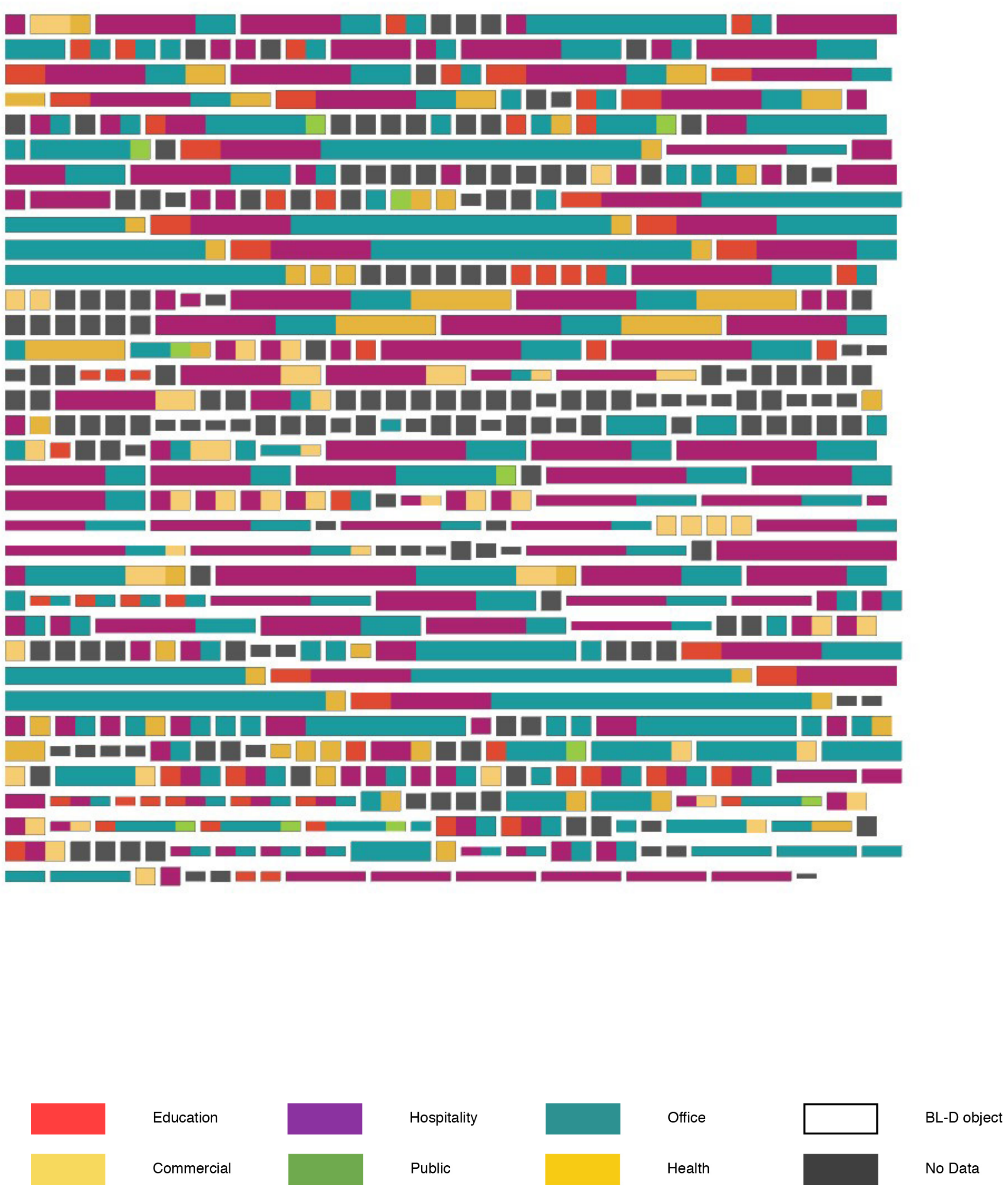
🟦 Project Site

⬛ No Data

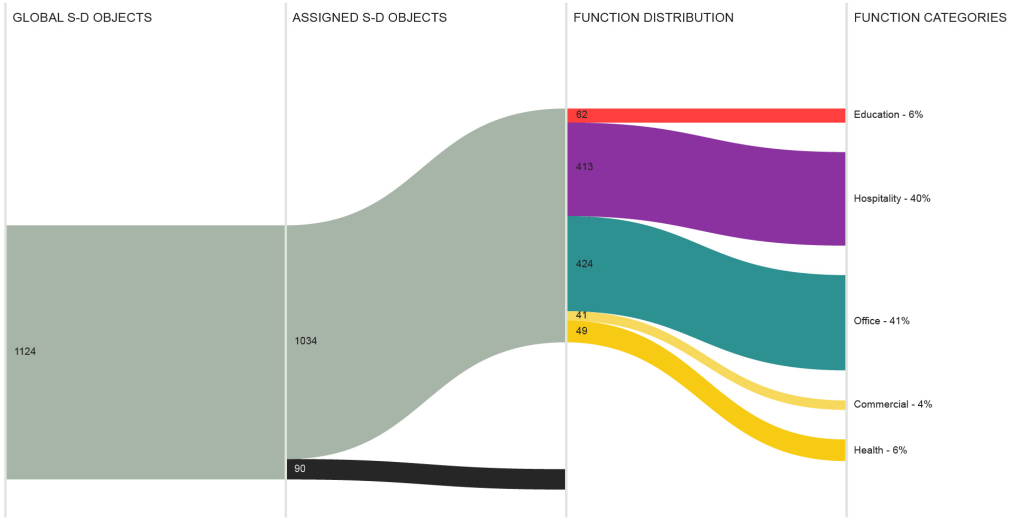
Data availability rate of 59%

6 PM

Time View



String height represents relative mesh volume of parent BL-D object


GLOBAL S-D OBJECTS
ASSIGNED S-D OBJECTS
FUNCTION DISTRIBUTION
FUNCTION CATEGORIES
1124
1034
90
62
413
424
41
49
Education - 6%
Hospitality - 40%
Office - 41%
Commercial - 4%
Health - 6%