Data Driven Parametric City Generation

MSc Geomatics P2 Thesis Proposal

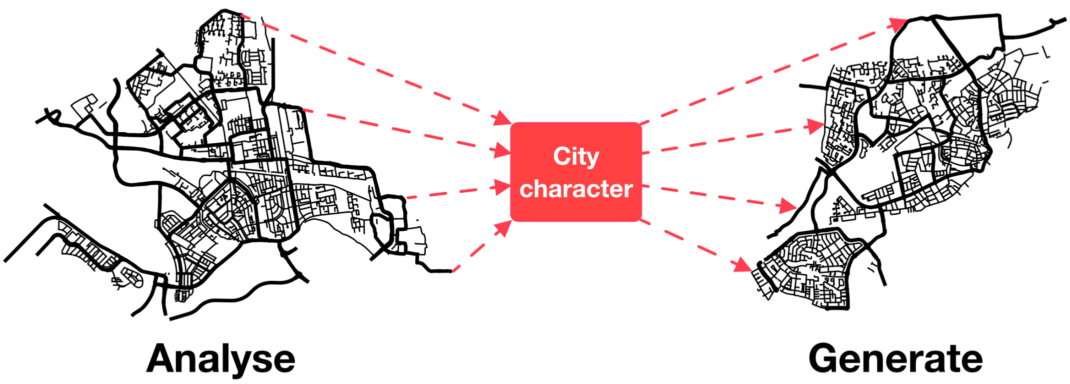
Oliver J. Post

Delft University of Technology

Supervisors:

Hugo Ledoux
Delft University of Technology

Akshay Patil
Delft University of Technology



Contents

1 Introduction

This is a thesis about cities that don't exist. This doesn't mean fantasy cities from an alternate universe or sci-fi cities from the future. Cities that don't exist, but look like they could exist. This topic has fascinated me personally for a long time. Starting from age 12, I was building cities in the game *Cities XL*, which I later replaced with the newer *Cities: Skylines*. I always found it very difficult to make a city look realistic, especially the road network. Even the apparent simplicity of the North American grid seemed impossible to recreate without something feeling "off". With this thesis, I want to give this complex but fun challenge another try from a academic angle.

Generating cities in 3D is not just constrained to city building games. In the field of Geomatics, 3D city models are an important form of urban data that offer a way to integrate the many domains of a real-life city, including terrain, buildings, vegetation, and road networks. These models combine both geometry and semantics in a single framework. Increasingly, 3D city models are used in more and more research domains, among which estimation of solar irradiation, energy demand estimation, estimation of the propagation of noise, and computational fluid dynamics (CFD) for wind simulation [1].

Despite their growing application, the creation and availability of accurate 3D city models present significant challenges. These include the limited availability of existing models [2], frequent geometric validity errors [3], and the intricate, labor-intensive process of creating detailed models [2]. While fully automatic generation using LIDAR data is feasible, it relies heavily on data availability and can still result in invalid geometries [4].

Parametrically generated city models propose a promising solution for numerous use cases. While many research topics need data on cities that exist in real life, a wide range of domains can utilize or even benefit from models of non-existing cities that offer parametric control over their creation. Implementing a parametric generator allows for the rapid production of new city models [5] and the easy application of city-wide changes, such as adjustments to building setback rules, which would be cumbersome manually.

Parametrically generated cities have diverse research and practical applications including urban and social simulations, serving as test files for algorithms, file formats and software, as input for parameter dimensionality reduction for Computational Fluid Dynamics (CFD) studies, and as training data for machine learning models. Other applied use cases for these models include virtual reality, video games, animation, rendering, and as tools in urban and architectural design prototyping. It is important to clarify that these procedural cities are not intended as designs for new cities, which would encroach on the roles of architects and urban planners. Instead, they are realistic variations of existing cities, to serve as tools in research, development and visualization.

Besides these use cases for parametric cities, Kim et al. [6] suggest that many research domains traditionally reliant on actual urban data could benefit from procedurally generated models. One advantage is the elimination of quality issues associated with volunteered geographic information (VGI). Furthermore, population data in these models can be detailed yet fully anonymous, avoiding privacy issues linked with real-world data by synthesizing cities that resemble, but are not directly connected to, real locations, buildings, or people.

Another potential application is the testing and public demonstration of urban plans that might be controversial, such as the placement of wind turbines. Using an existing city model for such tests could provoke local opposition if residents identify their surroundings in the test scenarios. A procedurally generated city, similar but not identical to the real one, allows for comprehensive planning and testing without the risk of controversy.

There are numerous existing methods for parametrically generating cities or their components, see Section 2.2. However, there are problems, which have to be addressed for the models to serve their purpose. Yet, these methods often lack plausibility [5], [6] as they are often based on generalized rule-based approaches instead of real city data. There is also a lack of availability of ready-to-use solutions that integrate the full range of city generation (layout, roads, buildings, etc.). Current options include theoretical models without publicly available code, expensive commercial software [7], and others that are limited in scope and realism.

This thesis proposes the development of a data-driven procedural city generator designed to create plausible, standardized, and valid 3D city models representative of various global city typologies. An extendable and modular approach will be employed to support the broad spectrum of possible applications.

2 Related Work

2.1 The City

To generate plausible cities, a deep understanding of urban structure is crucial. Urban morphology, the study of the physical form of cities, provides a robust theoretical foundation for this. Araújo De Oliveira, in *"Urban Morphology: An Introduction to the Study of the Physical Form of Cities"* [8], explains that urban morphology has diverse definitions. Key descriptions include:

> [Urban morphology is] an approach to conceptualizing the complexity of physical form. Understanding the physical complexities of various scales, from individual buildings, plots, street blocks, and the street patterns that make up the structure of towns helps us to understand how towns have grown and developed.
>
> – Larkham [9]

> A method of analysis which is basic to find(ing) out principles or rules of urban design.
>
> – Gebauer & Samuels [10]

Araújo De Oliveira states that cities are, morphologically, extremely complex objects. To be able to deal with these complexities, the city is divided into distinct elements of different scales that can be isolated from their context. He mentions from small to large scale: the natural context, the streets system, the plots system, and the buildings system. In this thesis these are referred to as the *scale layers* of the city, see Figure 1.

Morphology researcher Karl Kropf defines Urban Tissue as the character of an urban environment, which emerges from the interplay of the scale layers of the city [11]. The nature of urban tissue is dependent on the "resolution", the scale we are looking at the city. At a high resolution, the materials of individual buildings contribute among other factors towards the character of the city, while at a low resolution, the urban tissue only includes street patterns and block shapes [8]. Urban tissue not only varies from city to city but also within different zones of the same city.

This means the different scale layers can be analyzed separately and then combined to define the character of the city in the form of the urban tissue. Since the urban tissue can differ within the city, this analysis is not city-wide but in terms of zones within the city.

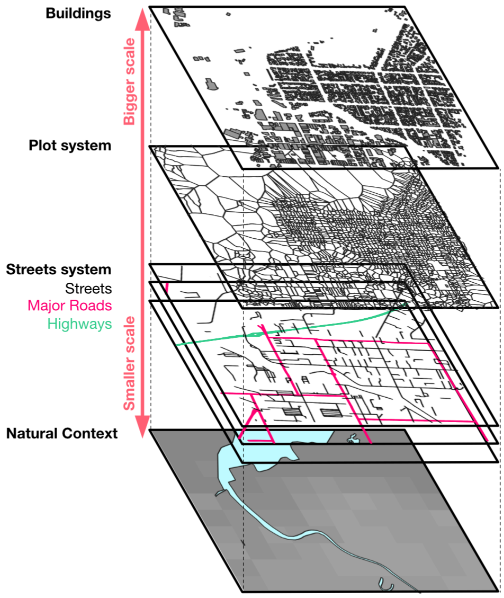


Figure 1: The scale layers of the city. Background data from OpenStreetMap [12] and GHSL [13].

2.2 City Generation

Many methods for generating (parts of) cities exist, and for clarity, these can be categorized. Kim et al. suggest categorizing procedural city generation techniques into generative grammar, simulation-based, tensor field, stochastic, data-driven, and inverse procedural generation [6]. Conversely, Smelik et al. propose a slightly different classification that includes artificial intelligence and computational geometry but excludes inverse procedural generation and tensor fields [14]. This thesis utilizes a hybrid of these classifications to explore the range of available methods.

The following is a summary of notable techniques from each category.

2.2.1 Generative Grammar

Shape grammars consist of rules that define the structure of a language, which can be used to generate geometry. For instance, Lindenmayer systems (L-Systems) were initially developed to model organic structures [15]. Parish & Müller adapted this concept to generate road networks from input image maps such as terrain and population density [16]. The process involves constructing the road network segment by segment, where the initial angle of each new segment is influenced by global goals. Adjustments like pruning, rotation, and snapping are applied to ensure each segment conforms to local constraints. Originally this L-system uses string reformatting, but Barett describes a way to efficiently implement this as an algorithm based on a priority queue [17]. This system produces plausible results and offers the possibility of extending it by modifying the global goals and local constraints.

In addition to street generation, generative grammars are widely used in procedural building generation systems [14]. One basic approach involves combining extruded primitives of varying heights [18]. Another technique employs L-systems on a rectangular floorplan, facilitating automatic Level of Detail (LOD) generation [16]. Müller et al. developed the Computer Generated Architecture (CGA) shape grammar to produce more intricate geometries [19].

The commercial software CityEngine integrates L-systems for street layout and CGA for building generation [7]. The major downside of shape grammars are their complexity and the required expertise to use them [20].

2.2.2 Simulation-Based

In *Interactive Geometric Simulation of 4D Cities*, the L-System is combined with traffic simulation to model city growth over time [21]. This integration yields plausible outcomes, though scaling to larger cities remains a challenge. *Procedural Modeling of Urban Land Use* employs agent-based simulation to determine land use within a raster representation of a city domain [22]. The land use patterns generated appear realistic; however, drawbacks include significant computation time and reliance on arbitrary rule sets based on assumptions rather than empirical data.

2.2.3 Tensor Field

Chen et al. introduce tensor fields as an intuitive and flexible framework for generating the streets system [23]. They argue that street patterns mostly consist of two dominant directions and they utilize the properties of tensor fields to generate major and minor roads from these mathematically derived directions. A big advantage of this method is the ease of use, as the patterns of the tensor field are reasonably intuitive and easy to manually edit. The results look plausible for North American gridiron city centers but are limited in plausibility for other types of urban fabric.

2.2.4 Stochastic

Purely stochastic generation methods are primarily used for random terrain generation [6], [14], which falls outside the scope of this thesis.

2.2.5 Data-Driven

Aliaga & Vanegas describe a technique that uses an existing road network as a template to generate extended road patterns while maintaining a similar structure [24]. Nishida et al. further this approach with their *grow*, *warp*, and *blend* operations [25]. These techniques generate plausible results for extending an existing example but currently do not seem suitable for producing plausible city-wide street patterns.

2.2.6 Inverse Procedural Generation

Inverse procedural generation addresses the challenge of complex input parameters in traditional procedural generation, which often require specialized knowledge and experimentation. This method determines optimal procedural inputs based on predefined goals or examples. Vanegas at al. introduce such a method for city generation. One example within this work is the user defined target of optimised sun exposure, where the model figures out the optimal procedural parameters to achieve this [26]. The main advantage of this method is that a procedural model can be reused to optimise for new target indicators without having to reprogram the original model.

2.2.7 Artificial Intelligence

Artificial intelligence applications in city generation include techniques like generative adversarial networks for road layouts [27] and deep learning for procedural parameter control [28]. However, "black box" methods are outside the scope of this thesis, and therefore, these techniques have not been extensively reviewed.

2.2.8 Computational Geometry

A variety of geometric methods are employed for parcel generation. Generating a Voronoi diagram based on points that represent building locations typically yields unconvincing results. Kelly introduces a straight-skeleton method for creating *modernist* parcel divisions typical of planned suburbs [29], which produces highly plausible outcomes but is specific to this parcel style. For other parcel types, Kelly suggests using Object Oriented Bounding Box (OOBB) methods. Vanegas et al. adapt Parish & Müller's parcel generation technique by recursively splitting the OOBB until a user-specified criterion is met [30], [16]. An alternative method is proposed by Emilien et al. [31] and uses a multistep process of anisotropic land conquest. This method was developed for generating rural landscapes but seems promising for urban landscapes. It has the ability to follow predetermined rules and costs.

2.3 City Analysis

Several studies have applied clustering to identify urban form typologies based on morphological metrics. Badhrudeen et al. [32] and Louf & Barthelemy [33] have used clustering on a global scale, aggregating metrics from individual roads within a city to articulate street pattern characteristics. Although capable of distinguishing between cities, this method does not differentiate between areas within the same city. Fleischmann et al. perform an alternative clustering on individual buildings with metrics to morphologically describe their direct neighborhood [34]. This enables the identification of distinct morphological areas within cities.

3 Research Objectives

3.1 Objectives

The main research question of this thesis is:

How can digital city models be procedurally generated to resemble the character of real-life cities and city archetypes?

The "character" of a city is defined as the composite of various urban tissue areas comprising the city and their spatial interrelationships.

This question is broken down into the following sub-questions:

- *How can the urban form of real-world cities be captured using publicly available geospatial data?*
- *What statistical conclusions can be drawn about the urban form of individual cities from this geospatial data?*
- *How can urban form archetypes be derived from these statistical conclusions?*
- *How can these statistical conclusions be utilized to procedurally generate a digital city model that resembles the form of a real-life city or urban form archetype?*

The anticipated outcomes of this thesis include:

- A command line tool for morphological analysis of real-life cities.
- A dataset containing simplified/aggregated results of this analysis.
- A set of urban tissue typologies, categorized by scale layer.
- A set of city character descriptions for cities worldwide, derived from combined urban tissue typologies.
- A tool for procedurally generating 3D city models based on a city character description and simple user inputs.

3.2 Scope

Cities are endlessly complex structures, from their general structure to the individual bricks of a building. This thesis deliberately narrows its scope to manage both the analytical and generative processes effectively.

This thesis:

- is restricted to 2.5D analysis, simplifying the complexity and enhancing data availability.
- excludes the procedural generation of the natural context, using existing terrain and water maps as starting point.
- aims to create plausible and realistic urban fabric without striving for photorealistic or highly detailed geometric models.
- avoids methods that produce results through "black box" processes, specifically excluding the use of deep learning techniques.

4 Methodology

The term urban tissue [11] can be used to describe the distinct character of city areas, encompassing various scale layers including terrain context, street systems, plot systems, and buildings. From this, we can state the character of an entire city to be the combination of its urban tissue areas, and their spatial interrelation.

To create cities with plausible character, it is essential to analyse the urban tissues of existing cities. The methodology of this thesis involves initially analyzing the urban tissue areas of cities worldwide. This process includes identifying distinct areas within each scale layer of the city. These areas are then aggregated to form a comprehensive "character template" for the city. This template serves as a foundation for generating new procedural cities that mirror the original character, as illustrated in Figure 2. Additionally, the characters of individual cities may be amalgamated to form generalized character archetypes for cities with similar traits.

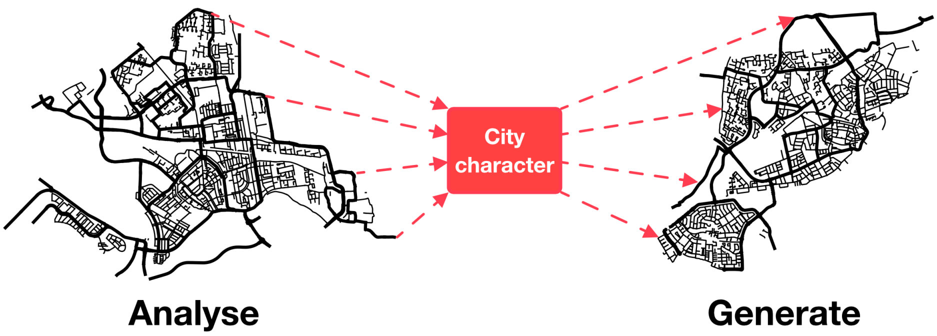


Figure 2: Conceptual diagram showing the pipeline from analysis of real-life cities into the city's "character template", which is then used to generate new procedural cities.

4.1 Analysis Methodology

This thesis describes urban characteristics using numerical metrics for each element within a city's scale layers. Elements in natural contexts consist of gridded spatial partitions. For street systems, elements are individual street segments. Plots and buildings are defined by their polygon geometries.

The analysis is structured into distinct stages, see Figure 3.

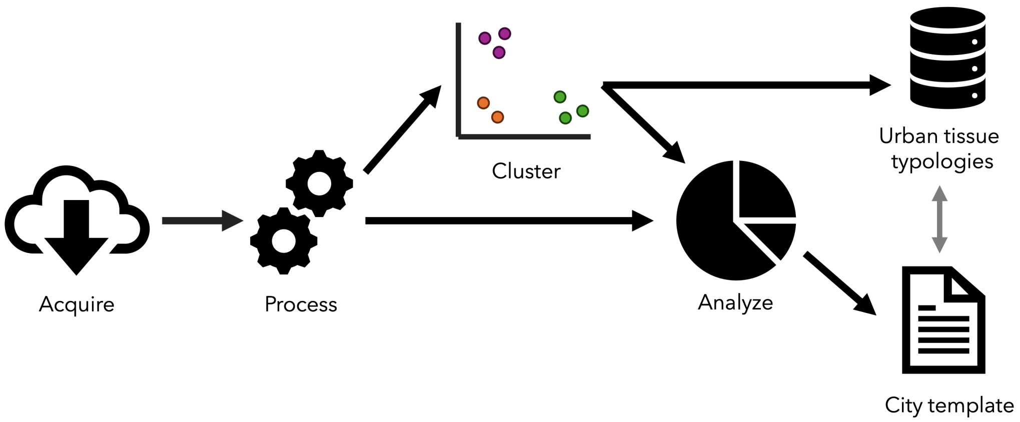


Figure 3: Analysis pipeline divided into stages, with as output global typologies and individual city character templates.

4.1.1 Data Acquisition

Initial elements and attributes are sourced and harmonised during the *acquire* stage from the sources mentioned in Section 6.2. Due to the absence of a global source for plot data, these elements are inferred from other scale layers in a subsequent stage.

4.1.2 Scale Layer Processing

The *process* stage enriches the data by creating the plots scale layer, adding supplementary layers, and computing morphological metrics for all layers. Metrics from existing libraries [35], [36] are mixed with newly developped metrics.

4.1.3 Clustering

The *cluster* stage categorises similar elements into distinct clusters to enhance understandability and comparability. By performing this clustering on the combined data of all analyzed cities, these clusters will be valid across cities. Clustering is based on a selection of metrics from each scale layer, and the resulting typology clusters are stored in a universal database valid for cities worldwide. Detailed statistics of these typologies, including metrics not directly used in clustering, are saved along with each typology.

4.1.4 Typology Grid

The aim of the *analyze* stage is to summarize the urban tissue of the city based on the results of the *cluster* stage. This is done by dividing each city scale layer into zones that consist of the same cluster, these are the distinct areas mentioned in the introduction of Section 4. Since most layers consist of distinct objects (i.e. road segments), a division of the scale layer into zones has to be made based on a planar partition of the city. To achieve results that allow for inter-city comparison, a regular-sized gridded planar partition is used, where each cell contains its most prominent cluster, see Figure 4. Distinct areas are formed by connected cells of the same cluster. This planar partition is referred to as the *typology grid*. The city character template is exported containing a description of the resulting typology grid of each scale layer. This statistrical description is based on cluster adjacencies, distance measures, correlation with clusters in scale layers of smaller scale, and patch-based statistics based on methodology from FRAGSTATS [37].

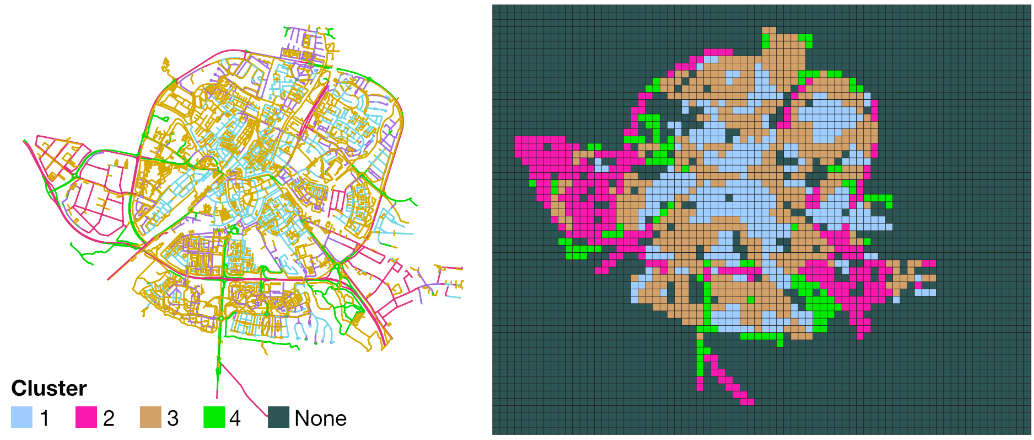


Figure 4: Experimental clustering of the street system scale layer (left) and the resulting typology grid (right).

4.2 Generation Methodology

Generation begins by establishing the natural context for the city to be generated. Since natural context generation is out of scope for this thesis, context may be derived from user-supplied input files or a user-defined real-world area, from which natural context maps are downloaded. Satellite acquired height maps [38] and VGI vector data [12] will be used as sources. The user then selects a limited set of parameters, including a template reflecting a specific city's character or a broader character archetype.

4.2.1 Typology Grid Generation

To generate a plausible city character, the spatial distribution of the urban tissue has to resemble that of the character template. The typology grid from the analysis phase (see Section 4.1.4) is used for this. The aim is to generate a new typology grid in the new context, where the layers are generated from top to bottom, and they have a similar distribution of typologies as the template city. This process starts with filling the layer of the typology grid with the typologies from the chosen city or archetype. The proportion of each typology is calculated based on a ratio linking the typology distribution in the template to user-defined parameters, such as city population. Initial placement employs a statistical heuristic method, followed by optimization through simulated annealing [39] to refine the layout. The plausibility of the layout is evaluated using an objective function that rewards statistical similarity to the template. This similarity assessment includes type adjacency, patch-based metrics [37], and alignment with lower scale layers of the typology grid, ensuring, for example, that road patterns are consistent with the terrain scale layer. The next layer can only be generated when the layer below it is finished. This limits complexity, while retaining the interrelationships of the scale layers of the city.

4.2.2 Streets System Generation Methodology

Street system generation employs Barett's adaptation [17] of the L-Systems method by Parish & Müller [16]. A prototype has been successfully implemented in Python to validate the method; final implementation will be in Rust.

This system's global constraints have been refined to enhance the plausibility of city models and to integrate data from the analysis phase. Local constraints, including snapping, truncation, and deflection of streets, remain unchanged.

The natural context continues to serve as user input. However, the population map used in the original implementation is replaced by the *typology grid*. This grid assigns street pattern typologies from the analysis phase to specific city locations, influencing the angle and length of new street segments based on the statistical character of the typologies, rather than the original population-driven approach.

Each road type—highways, major roads, streets—is handled within its own scale sublayer in the typology grid, as these are clustered separately during the analysis phase. This means the typology grid of lower-level roads is influenced by that of higher-level roads.

4.2.3 Plot System Generation Methodology

Prototype development of the existing methods mentioned in Section 2.2.8 will determine the chosen method. The land conquest method from Emilien et al. [31] is preferable since it can work for all types of plots. The building seeds that the plots are generated from would be placed within the confines of each block, based on distribution statistics from the plot typology. Constraints and costs for the land conquest algorithm would also be derived from statistics from the plot typology.

4.2.4 Building Generation Methodology

This thesis does not focus on generating detailed building geometries. Instead, it prioritizes replicating essential characteristics that contribute to plausible urban tissue, such as whether buildings are free-standing or connected. The method involves generating 2D building footprints using a simplified shape grammar, followed by extrusion to heights statistically determined from the character template.

4.3 Verification

The final stage involves statistically verifying the generated city's quality and plausibility. Verification will include expert assessments and statistical comparisons between the morphological metrics of the created city and the template city.

5 Schedule

5.1 Activities

The analysis and development process is highly complex, as it involves a back-and-forth between analysis and generation and between the different city scale layers. Any part can be made infinitely complex, due to the complexity of real cities, and therefore there is no way to determine if a part is finished. This makes traditional waterfall planning methods, where the project is divided into phases with start and end dates, unsuitable for this project. Instead, the choice has been made to utilize a planning method according to Agile principles [40], specifically an adapted version of the Scrum [41] framework.

This involves creating a list of all work that "could" be done, including all feature ideas. This is called the "backlog". These items are sorted based on priority, where essential features are at the top. After each meeting, a set of high-priority items from this list is chosen to be worked on until the next meeting. This time period is called the "sprint". The idea is to work towards a barebones but functioning version of the whole pipeline first, prospected to be finished in July, and then in every additional sprint to add a set of new features and end up again with a fully functioning but improved pipeline at the end of the sprint. This means that at any moment a fully functional pipeline exists, so there is no possibility of ending up with no results or having to rush at the end to produce something functional. This process allows for iterative feedback and results. Each sprint will also include writing, so the thesis document is iteratively constructed and can be refined along the way. I believe this system is preferable to predefined time planning since it allows for a high amount of flexibility while still guaranteeing final results.

To facilitate this process, the software Jira [42] is used, which provides a powerful set of tools for the Scrum workflow. The tasks (called "issues") are divided into categories (called "epics") that make sure the different areas of the thesis get an even amount of progress.

5.2 Meetings

To accommodate for iterative feedback, weekly/bi-weekly meetings are held with the supervisors Hugo Ledoux and Akshay Patil. During these meetings, the progress of the work is presented followed by feedback and discussion. To get feedback on the urbanism perspective, a meeting was held with Claudiu Forgaci - Assistant Professor of Urban Design - and a future meeting is likely. Additionally, a presentation was held to present the current progress to the 3D Geoinformation Group, where feedback was gathered on the technical approach.

6 Tools and Data

6.1 Tools

Data analysis will be performed in the Python [43] programming language . The libraries `osmnx` [35] and `momepy` [36] are used for data acquisition and processing. These libraries are based on `geopandas` dataframes [44] and `networkx` graphs [45].

The code for city generation is aimed to be written in the Rust [46] programming language due to it's superior performance and easy interfacing with Python using Maturin and PyO3 [47].

QGIS [48] is used for the visualization and analysis of final and intermediate datasets.

To be able to process hundreds of cities worldwide, a 32-core 256 GB RAM Linux server of the TU Delft 3D Geoinformation group [49] is used in combination with GNU parallel [50] to run separate Python processes on multiple cores.

6.2 Data

The amount of urban data out there is endless, therefore it is important to select what specific data sources will be used. Since this thesis aims to generate plausible cities worldwide, it is of high importance that the selected data sources offer (near) worldwide coverage. Below is an overview of the data sources that have been implemented already and the ones that have not yet been implemented but are perceived of high value.

All features from OpenStreetMap are downloaded using the `osmnx` library [35]. A custom instance of the Overpass API [51] has been set up on a local server to not be constrained by the rate limits of the public API.

6.2.1 Natural Context

- `Implemented` Vector water features from OpenStreetMap [12] features with `natural` tag with the values `water`, `coastline`, `bay` or `strait`.
- `Proposed` Terrain height data (DSM) from SRTM [38]. Optimal would be DTM data, but no openly available DTM with worldwide coverage has been identified yet.

6.2.2 Streets System

- `Implemented` Road network line features from OpenStreetMap [12], as identified by the `highway` tag.

6.2.3 Plots System

- `Implemented` Enclosed morphological tessellation based on buildings, generated using Momepy [36].

6.2.4 Buildings

- `Implemented` Building footprints from OpenStreetMap [12] elements with `building` tag.
- `Proposed` Higher coverage building footprint dataset from the Overture Maps Foundation [52].
- `Partially Implemented` Building height from Global Human Settlement GHS-BUILT-H dataset [13].
- `Proposed` Building height for individual buildings from a database of tall buildings. Several of these databases have been identified [53]–[55], but none of the identified sources offers a publicly available API or any form of bulk data export.

6.2.5 Supplementary Data

- `Implemented` Land-use and airport regions from OpenStreetMap [12].
- `Implemented` Administrative border and city center from OpenStreetMap [12].
- `Proposed` Population data from Meta [56] or GHSL [57].

References

[1] F. Biljecki, J. Stoter, H. Ledoux, S. Zlatanova, and A. Çöltekin, "Applications of 3D City Models: State of the Art Review," *ISPRS International Journal of Geo-Information*, vol. 4, no. 4, pp. 2842–2889, Dec. 2015, doi: 10.3390/ijgi4042842.

[2] R. Girindran, D. Boyd, J. Rosser, D. Vijayan, G. Long, and D. Robinson, "On the Reliable Generation of 3D City Models from Open Data," *Urban Science*, 2020, doi: 10.3390/URBANSCI4040047.

[3] H. Ledoux, "Val3dity: Validation of 3D GIS Primitives According to the International Standards," *Open Geospatial Data, Software and Standards*, vol. 3, no. 1, p. 1–2, Dec. 2018, doi: 10.1186/s40965-018-0043-x.

[4] R. Peters, B. Dukai, S. Vitalis, J. van Liempt, and J. Stoter, "Automated 3D Reconstruction of LoD2 and LoD1 Models for All 10 Million Buildings of the Netherlands," vol. 88, no. 3. pp. 165–170, 2022. doi: 10.14358/PERS.21-00032R2.

[5] S. A. Groenewegen and R. M. Smelik, "Procedural City Layout Generation Based on Urban Land Use Models."

[6] J.-S. Kim, H. Kavak, and A. Crooks, "Procedural City Generation beyond Game Development," *SIGSPATIAL Special*, vol. 10, no. 2, pp. 34–41, Nov. 2018, doi: 10.1145/3292390.3292397.

[7] "CityEngine." [Online]. Available: https://www.esri.com/en-us/arcgis/products/arcgis-cityengine/overview

[8] V. M. Araújo De Oliveira, *Urban Morphology: An Introduction to the Study of the Physical Form of Cities*. in The Urban Book Series. Cham: Springer International Publishing, 2022. doi: 10.1007/978-3-030-92454-6.

[9] P. Larkham, "Understanding Urban Form?," *Urban Design*, no. 93, pp. 22–24, 2005.

[10] M. Gebauer and I. Samuels, "Urban Morphology: An Introduction.," *Joint Centre for Urban Design, Research Note 8. Oxford Polytechnic, Oxford*, 1981.

[11] s. Kropf, "Urban Tissue and the Character of Towns," *Urban Design International*, vol. 1, pp. 247–263, Sep. 1996, doi: 10.1057/udi.1996.32.

[12] OpenStreetMap contributors, "Planet Dump Retrieved from https://planet.osm.org." [Online]. Available: https://www.openstreetmap.org/

[13] M. Pesaresi, "GHS-BUILT-H R2023A - GHS Building Height, Derived from AW3D30, SRTM30, and Sentinel2 Composite (2018)." Euro-

pean Commission, Joint Research Centre (JRC), Apr. 25, 2023. doi: 10.2905/85005901-3A49-48DD-9D19-6261354F56FE.

[14] R. Smelik, T. Tutenel, R. Bidarra, and B. Benes, "A Survey on Procedural Modeling for Virtual Worlds," 2014. Accessed: Nov. 28, 2023. [Online]. Available: https://www.semanticscholar.org/paper/A-Survey-on-Procedural-Modeling-for-Virtual-Worlds-Smelik-Tutenel/05311671fe92fc09fdc153c14654a16fb63b7348

[15] P. Prusinkiewicz and A. Lindenmayer, *The Algorithmic Beauty of Plants*, First soft cover printing. in The Virtual Laboratory. New York Heidelberg: Springer, 1996.

[16] Y. I. H. Parish and P. Müller, "Procedural Modeling of Cities," doi: 10.1145/1185657.1185716.

[17] S. Barett, "L-Systems Considered Harmful." Accessed: Dec. 16, 2023. [Online]. Available: http://nothings.org/gamedev/l_systems.html

[18] S. Greuter, J. Parker, N. Stewart, and G. Leach, "Real-Time Procedural Generation of `pseudo Infinite' Cities," presented at the Proc. GRAPHITE, Feb. 2003, pp. 87–94. doi: 10.1145/604471.604490.

[19] P. Müller, P. Wonka, S. Haegler, A. Ulmer, and L. Van Gool, "Procedural Modeling of Buildings," *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 614–623, Jul. 2006, doi: 10.1145/1141911.1141931.

[20] R. M. Smelik, "A Declarative Approach to Procedural Generation of Virtual Worlds."

[21] B. Weber, P. Müller, P. Wonka, and M. Gross, "Interactive Geometric Simulation of 4D Cities," *Computer Graphics Forum*, vol. 28, no. 2, pp. 481–492, Apr. 2009, doi: 10.1111/j.1467-8659.2009.01387.x.

[22] T. Lechner, P. Ren, B. Watson, C. Brozefski, and U. Wilenski, "Procedural Modeling of Urban Land Use," in *ACM SIGGRAPH 2006 Research Posters on - SIGGRAPH '06*, Boston, Massachusetts: ACM Press, 2006, p. 135–136. doi: 10.1145/1179622.1179778.

[23] G. Chen, G. Esch, P. Wonka, P. Muller, and E. Zhang, "Interactive Procedural Street Modeling."

[24] D. G. Aliaga and C. A. Vanegas, "Interactive Example-Based Urban Layout Synthesis."

[25] G. Nishida, I. Garcia-Dorado, and D. G. Aliaga, "Example-Driven Procedural Urban Roads," *Computer Graphics Forum*, vol. 35, no. 6, pp. 5–17, Sep. 2016, doi: 10.1111/cgf.12728.

[26] C. A. Vanegas, I. Garcia-Dorado, D. G. Aliaga, B. Benes, and P. Waddell, "Inverse Design of Urban Procedural Models," *ACM Transactions on Graphics*, vol. 31, no. 6, pp. 1–11, Nov. 2012, doi: 10.1145/2366145.2366187.

[27] S. Hartmann, M. Weinmann, R. Wessel, and R. Klein, "StreetGAN: Towards Road Network Synthesis with Generative Adversarial Networks."

[28] D. Ritchie, A. Thomas, P. Hanrahan, and N. D. Goodman, "Neurally-Guided Procedural Models: Amortized Inference for Procedural Graphics Programs Using Neural Networks." Accessed: Jun. 06, 2024. [Online]. Available: http://arxiv.org/abs/1603.06143

[29] T. Kelly, "Unwritten Procedural Modeling with the Straight Skeleton."

[30] C. A. Vanegas, T. Kelly, B. Weber, J. Halatsch, D. G. Aliaga, and P. Müller, "Procedural Generation of Parcels in Urban Modeling," *Computer Graphics Forum*, vol. 31, no. 2pt3, pp. 681–690, May 2012, doi: 10.1111/j.1467-8659.2012.03047.x.

[31] A. Emilien, A. Bernhardt, A. Peytavie, M.-P. Cani, and E. Galin, "Procedural Generation of Villages on Arbitrary Terrains," *The Visual Computer*, vol. 28, Jun. 2012, doi: 10.1007/s00371-012-0699-7.

[32] M. Badhrudeen, S. Derrible, T. Verma, A. Kermanshah, and A. Furno, "A Geometric Classification of World Urban Road Networks," *Urban Science*, vol. 6, no. 1, p. 11–12, Feb. 2022, doi: 10.3390/urbansci6010011.

[33] R. Louf and M. Barthelemy, "A Typology of Street Patterns." Accessed: Jan. 21, 2024. [Online]. Available: http://arxiv.org/abs/1410.2094

[34] M. Fleischmann, A. Feliciotti, O. Romice, and S. Porta, "Methodological Foundation of a Numerical Taxonomy of Urban Form," *Environment and Planning B: Urban Analytics and City Science*, vol. 49, no. 4, pp. 1283–1299, May 2022, doi: 10.1177/23998083211059835.

[35] G. Boeing, "OSMnx." Accessed: Aug. 03, 2023. [Online]. Available: https://github.com/gboeing/osmnx

[36] "Momepy: Toolkit for Urban Morphology." [Online]. Available: https://github.com/martinfleis/momepy

[37] K. McGarigal and B. J. Marks, "FRAGSTATS: Spatial Pattern Analysis Program for Quantifying Landscape Structure.," 1995. doi: 10.2737/PNW-GTR-351.

[38] OpenTopography, "Shuttle Radar Topography Mission (SRTM) Global." Accessed: Jun. 04, 2024. [Online]. Available: https://opentopography.org/meta/OT.042013.4326.1

[39] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by Simulated Annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, May 1983, doi: 10.1126/science.220.4598.671.

[40] K. Beck *et al.*, “Manifesto for Agile Software Development.” [Online]. Available: https://agilemanifesto.org/

[41] Ken Schwaber and Jeff Sutherland, “The Scrum Guide.” Accessed: Jun. 06, 2024. [Online]. Available: https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf#zoom=100

[42] Atlassian, “Jira | Issue & Project Tracking Software | Atlassian.” Accessed: Jun. 07, 2024. [Online]. Available: https://www.atlassian.com/software/jira

[43] G. Van Rossum and F. L. Drake Jr, *Python Reference Manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.

[44] K. Jordahl *et al.*, “Geopandas/Geopandas: V0.8.1.” [Online]. Available: https://doi.org/10.5281/zenodo.3946761

[45] A. A. Hagberg, D. A. Schult, and P. J. Swart, “Exploring Network Structure, Dynamics, and Function Using NetworkX,” in *Proceedings of the 7th Python in Science Conference (SciPy2008)*, G. Varoquaux, T. Vaught, and J. Millman, Eds., Aug. 2008, pp. 11–15.

[46] N. D. Matsakis and F. S. Klock II, “The Rust Language,” in *ACM Sigada Ada Letters*, ACM, 2014, pp. 103–104.

[47] PyO3 Organisation, “PyO3.” Accessed: Jun. 05, 2024. [Online]. Available: https://github.com/pyo3

[48] QGIS Development Team, “QGIS Geographic Information System.” QGIS Association, 2024. [Online]. Available: https://www.qgis.org/

[49] “3D Geoinformation at TU Delft.” Accessed: Jun. 05, 2024. [Online]. Available: https://3d.bk.tudelft.nl/

[50] O. Tange, “GNU Parallel 20240422 ('Børsen').” Accessed: May 15, 2024. [Online]. Available: https://zenodo.org/doi/10.5281/zenodo.11043435

[51] Overpass, “Installation of an Own Instance.” Accessed: Jun. 05, 2024. [Online]. Available: https://dev.overpass-api.de/overpass-doc/en/more_info/setup.html

[52] Overture Maps, “Overture Maps Foundation Releases Its First World-Wide Open Map Dataset – Overture Maps Foundation.” Accessed: Nov. 28, 2023. [Online]. Available: https://overturemaps.org/overture-maps-foundation-releases-first-world-wide-open-map-dataset/

[53] "SKYDB – The World of Tall Buildings." Accessed: Jun. 07, 2024. [Online]. Available: https://www.skydb.net/

[54] "Database - SkyscraperPage.Com." Accessed: Jun. 07, 2024. [Online]. Available: https://skyscraperpage.com/cities/

[55] "The Skyscraper Center." Accessed: Jun. 07, 2024. [Online]. Available: https://www.skyscrapercenter.com/explore-data

[56] "Data For Good at Meta High Resolution Population Density Maps." Accessed: Nov. 28, 2023. [Online]. Available: https://dataforgood.facebook.com/dfg/tools/high-resolution-population-density-maps

[57] M. Schiavina, S. Freire, and K. MacManus, "GHS-POP R2023A - GHS Population Grid Multitemporal (1975-2030)." European Commission, Joint Research Centre (JRC), Apr. 25, 2023. doi: 10.2905/2FF68A52-5B5B-4A22-8F40-C41DA8332CFE.