FROM SKETCH TO DESIGN

A Cross-scale Workflow for Procedural Generative Urban Design

JIN GAO[1] and SAYJEL VIJAY PATEL[2]
[1] Massachusetts Institute of Technology.
[2] Digital Blue Foam
[1]gaojin@mit.edu, 0000-0002-1595-6895
[2]sayjel@digitalbluefoam.com

Abstract. This study aims to transform urban generative design by aligning it with the symbolic thinking of urban designers, creating an interactive web-based tool. Urban planners and designers use graphic and symbolic thinking. Yet, computer-aided design software accepts inputs by numeric parameters, while recent generative design tools focus on quantity rather than controllability, where in most cases there are "black boxes" hiding the algorithm and procedural logic behind the hood. As a result, there is an established need to tailor tools to the mindset of urban designers to make them more user-friendly and practical. To address this need, we develop a tool that takes the users' intuitive sketch as symbolic inputs to guide 3d urban form generation. It seamlessly combines symbolic thinking and urban form generation using ML-based gesture recognition middleware, and Tensor Field-based procedural urban network generation algorithm. The contribution of this work is: a review of urban generation algorithms and methods of human computation interaction, conceptual overview and methodology, and case study evaluation. We analyzed traditional and state-of-the-art urban design generative algorithms, such as cellular automata-based and machine learning image generation-based methods. It also compared different methods of human-computer interaction and selected appropriate urban design symbol systems. Combining both, the study develops a web-based interactive program backed by a Tensor Field-based procedural urban network generation algorithm, with JavaScript and React framework based on ML-based gesture recognition middleware. The research proposes a workflow for generative urban design. It further evaluates the possibility of integrating block or building scale, with generative architectural design platforms, to automatically refine and analyze rough urban volumes. Ultimately, this work points to a new future of tool development with tools that conform to the logic of urban growth while complying with the designer's mind.

Keywords. generative urban design, procedural modeling, urban semiotics, gesture recognition, tensor fields, creativity support tools

1. Introduction

Urban design emphasizes rational thinking with graphics. Among them, symbols, the abstract expressions of urban characteristics, are the fundamental language forming complex urban design proposals. The process of design is essentially a process of defining symbols. As a shared graphical language, they help urban designers understand and think simultaneously about multiple facets of urban attributes, including urban axis, density, form, function, etc. Urban theories also introduce semiotics as a way to understand the city. In the Image of City (Lynch, 1964), the conceptual "elements" are used to map the perception of cities, while Space, symbol, city (Zhu, 1993) categorizes cities as a collective entity of symbolic elements.

Generative urban modeling has been a long-term interest for planners, designers, and scholars. Due to the varied historical and planning backgrounds of each city, along with their complex and ever-changing nature, there is no universal methodology and tools for procedural urban modeling. The parameters controlling generation vary widely and may involve randomized parameters, user-input text boxes or sliders, or derived from GIS data, which is far from the actual graphical design thinking process. In addition, one set of input parameters often corresponds to multiple results, and it is often difficult for users to control the generation process. However, the designer's way of thinking is based on symbols and sketches, and a process of continuous adjustment based on visual feedback.

Currently, interactive urban design tools that can understand the symbolic language of designers are in the early stages of research. Designers use common architectural modeling workflows, including Rhino and Sketchup, to operate a large volume of masses. The workflow is unable to perform rapid modification and includes unnecessary details that are irrelevant to the core principles of urban design. As Humans are the primary decision-makers in the design process, there is also a lack of intuitive input and output methods that align with designers' symbolic language. These challenges highlight the need for tools that are more intuitive for designers and adaptable across different use cases.

The objective of this study is to propose a cross-platform generative urban design tool that simplifies urban design workflow. This tool aims to bridge the gap between core urban design methodologies and the intuitive, creative process of designers. The significance of this study lies in its potential to make urban design more accessible, efficient, and integrated with existing design workflows.

2. Related Works

2.1. VOCABULARY OF URBAN DESIGN

In both design education and practices, architectural and urban elements are usually interpreted as a "language" consisting of a vocabulary of patterns, as categorized in A Pattern Language (Alexander, 1977) and Concept Sourcebook (White, 1975). In discussions and collaborations, hand sketches and symbol languages are commonly used in all kinds of design processes.

2.2. GENERATIVE URBAN DESIGN

As computational design becomes widely used, scholars have tried to model the generation of urban environments from various perspectives. Learned from the self-organizing properties of urban systems, the Cellular Automata (CA) (Batty, 1997), and its derived SLEUTH model (slope, land use, exclusion, urban extent, transportation, and hillshade) (Triantakonstantis & Mountrakis, 2012) is widely used in regional scale modeling. Many generative models are based on transforming designers' urban planning and design logic into procedural algorithms. The literature (Beirao, Mendes, Duarte, & Stouffs, 2010, Mei, Pan, Cheng, & Garcia Del Castillo Lopez, 2021) encodes the urban planner's decision into parametric moves with which designers can generate organic or grid-like city models with assigned building height and programs based on common urban patterns. Scholars have also introduced architecture-scale computational design methods into urban scale. This includes the tools (Wilson, Danforth, Davila, & Harvey, 2019) to facilitate decision-making by generating correlated economic indicators, and the data-driven method (Olascoaga & Emilio, 2016, 2021) to model the city utilizing shape grammar. Recently, generative modeling methods, such as Machine Learning (ML), have been used in tasks including finding the figure-ground relationship to predict missing urban networks (Boim, Dortheimer, & Sprecher, 2022). However, compared with traditional methods that are based on logic, ML methods face the difficulties of extracting meaningful data from complex urban environments, the computation resources to train and implement models, and unpredictable outcomes.

The tensor-based method, applied in our research, was first proposed by (Chen, Esch, Wonka, Muller, & Zhang, 2008) and further improved and implemented with TypeScript by ProbableTrain (Keir, 2023), who produced an open-source, browser-based city generation program. This method abstracts the city network into a collection of tensors which represents the flow of common urban structures: grid and radius. Environmental factors, such as terrain and coastlines, can affect the tensor field thereby producing continuous results no matter the complexity of environmental constraints. On top of the city's basic grid, we can perform actions including classifying multiple levels of roads and merging city parcels. The advantage of this approach lies in its balance between the randomness of generation and the adherence to urban planning principles, facilitating the generation of cities that are both adaptive to context and well-structured as real-life cities.

This paper focuses on exam:

- How do symbols function as a fundamental language in complex urban design proposals, and how can they be defined and utilized in the design process?
- What are the challenges and variations in parameters controlling procedural urban modeling, and how can a generative urban design tool address them to align with the visual thinking process while providing efficient and adaptable solutions?
- What are the current limitations and challenges in existing interactive urban design tools, particularly in relation to their understanding of the symbolic language used by designers, and how can a cross-platform generative urban design tool bridge the gap between traditional urban design workflow and the creative designing process.

2.3. DESIGN TOOLS AND HUMAN-COMPUTER INTERACTION

Traditional urban design aid tools are mostly developed in the form of plugins set in commonly used professional software (Rhino, Sketchup, and ArcGIS). Exemplified works include the DeCodingSpaces Toolbox (Koenig, Miao, Bus, Knecht, & Chang, 2017) and the Urban Network Analysis Toolbox (Sevtsuk, 2021) as Rhino plugins, and the CityEngine (Kelly, 2021) as standalone desktop software based on the ArcGIS ecosystem. These tools have advanced analysis and generation capabilities, but their use requires dedicated training and professional knowledge. Their inability to cross-platform requires users to manually configure the running environment and rely on the host software for data exchange.

The pursuit to enhance the interactivity and ease of use of professional urban design tools has been a long-standing endeavor. Research groups represented by MIT Media Lab have proposed multiple forms of urban interactive modeling tools, such as the Augmented Urban Planning Workbench (Ishii et al., 2002), and CityScope (Alonso et al., 2018). These projects often feature a backend that simulates the built environment, including wind analysis, crowd dynamics, and traffic flows. Then, using an interactive frontend, such as projection combined with a touchable three-dimensional interface, to provide feedback on people's operations. With an emphasis on user interaction, these explorations enabled more dynamic and participatory approaches to urban design that foster a more inclusive and democratic urban planning process. The challenges they face are the sophisticated setup of hardware-software systems and limited use cases can make the technology less accessible.

Recently, more urban design tools have been developed using web technology. Thanks to the advances of WebGL and front-end frameworks, building cross-platform programs based on web pages is becoming more agile and efficient. Sophisticated functions, like advanced AI engines, can be implemented through high-performance servers running on the backend servers while maintaining lightweight, responsive user interfaces on the client side. These tools are often conveniently offered as Software as a service (SaaS), providing opportunities for business success. Representatives include Digital Blue Foam's Generative Design tool (DBF, n.d.), Delve (Delve by Sidewalk Labs - Real Estate Generative Design, n.d.), and KPF Urban Interface's Scout (Scout, n.d.). The emergence of web-based urban design tools represents an evolution in making them cross-platform, out-of-the-box accessibility, and SaaS-friendly.

3. Methodology

3.1. DEFINE GESTURE LANGUAGE SYSTEM FROM USER STUDY

To understand how designers utilize symbolic language in their sketches, we collected and analysed samples of urban design diagrams, and conducted experiments under a controlled setting by asking graduate-level urban design major students to draw interpretations of specific sites using their own symbolic language. We then categorized commonly used symbolic languages into basic shapes like stars, squares, and circles, to define a system of symbols. As samples listed in Figure 1, each of these symbols is meant to represent different types of influential parameters in urban design, including height, density, and program, which eventually shape the physical and functional aspects of the generated urban spaces.

Shape	Meaning	Shape	Meaning	Shape	Meaning
□	Grid Tensor Area	✚	High Density	╱	Main Axis
○	Radial Tensor Area	☆	Central Business District	∫	Custom (Curvy) Axis

Figure 1. Example of urban design symbolic language

3.2. CROSS-SCALE WORKFLOW

The research proposes a cross-scale workflow for generative urban design. As shown in Figure 2, the workflow takes the user's intuitive ideas and context information as input, generates urban design scale massing, and then zooms into a plot to perform architectural scale massing design and room division. Then, both the urban scale data: proximity, accessibility, porosity, FAR, and architectural scale data: floor area, structure, material usage, and energy efficiency, are evaluated through a comprehensive analysis process. Eventually, the generated project can be output as standard 3D models, allowing for refined rendering or detailed analysis with dedicated professional software, such as structural calculation and performance reports.

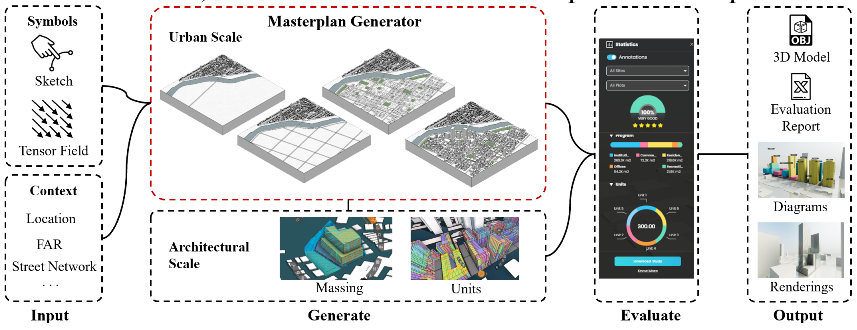

Figure 2. A Generative urban design workflow

3.3. PROTOTYPING OF THE TOOL

3.3.1. Software Architecture

The prototype is a web application that mainly features urban masterplan generation. Shown in Figure 3, it mainly consists of three components, a generation engine running at the backend, a render engine to visualize the generated models, and a frontend framework taking users' inputs. Between each component, the information is interchanged through inner APIs in the format of JSON objects.

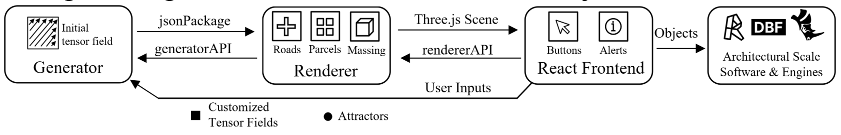

Figure 3. Software Architecture

3.3.2. Symbolic inputs and gesture recognition

The prototype featured symbol recognition of user sketches using the $1 Unistroke Recognizer Library (Wobbrock, Wilson, & Li, 2007). $1 represents the machine learning algorithm that uses an instance-based nearest-neighbor classifier. The hand sketches are translated into attractors that can be read by Tensor Field. Just like designing on paper, users can switch between tools such as pencils, erasers, and lines through the graphical buttons. Sketches will be automatically recognized and translated as symbols with their position and size. Furthermore, these symbolic languages serve as inputs to the multi-dimensional grid that stores calculated parameters to influence the generative process. Details will be explained in the next paragraph.

3.3.3. Urban Environment Generation

In our study, we explored state-of-the-art methods for generating urban networks and selected tensor-based methods due to their controllability, expandability, predictability, and high performance. Our prototype incorporates the urban generation module from the open-source program MapGenerator (Keir, 2023), which is based on the algorithm proposed by (Chen et al., 2008) and implemented in TypeScript. We further developed a control layer to take user inputs to influence the generation process. A multi-dimensional grid with nodes is used to record urban parameters. After receiving user input for tensor locations, and symbolic information, the program generates the 3D models in real time based on the tensor field and corresponding parameter grid.

Step 1 Environment Definition: Within a set rectangle boundary with a range of 1000m by 1000m (can be modified), the algorithm first randomly generates a coastline that defines the boundary between land and water. The boundary serves as the base input for the context of the urban layout and influences the overall structure of the tensor field, therefore influencing the generated urban network. The tensor field can be represented mathematically by (Chen et al., 2008):

$$R = \begin{bmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{bmatrix}$$

$where\ R \geq 0\ and\ \theta\ \in [0,2\pi).$

Step 2 Urban Network Generation: The street network is modeled as a graph G = (V, E), where V represents a set of nodes (intersections and endpoints) and E embodies a set of edges (streets). Initially, the site has a default grid tensor field. Users can introduce attractors by adding symbols. The attractors represent different characteristics of the base tensor field. Currently, the base tensor field has grid and radial types. The algorithm computes gradients between each tensor field defined by the attractor and updates dynamically. The grid field is defined as (Chen et al., 2008):

$$\ell = \sqrt{v_x^2 + v_y^2} \quad \text{and} \quad \theta = \arctan\left(\frac{v_y}{v_x}\right), \quad T(p) = e^{-d\|p-p_0\|^2} \begin{bmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{bmatrix}$$

For the radial pattern, the basis field is given by:

$$T(p) = e^{-d\|p-p_0\|^2} \begin{bmatrix} y^2 - x^2 & -2xy \\ -2xy & -(y^2 - x^2) \end{bmatrix},$$

$where\ x = x_p - x_0\ and\ y = y_p - y_0.$

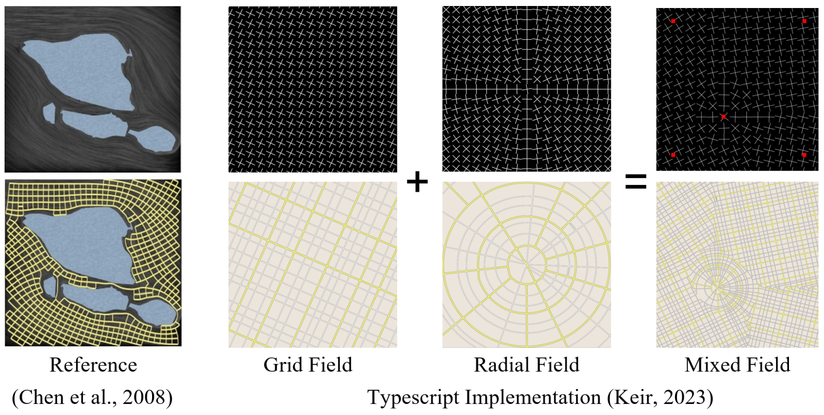

Figure 4. Transform the tensor field into street graphs (Chen et al., 2008)

Figure 4 describes how the algorithm finds street networks as streamlines within tensor fields. The street network is then divided into two hierarchies: major and minor roads. The hierarchy is defined by the distance and numbers between streamlines.

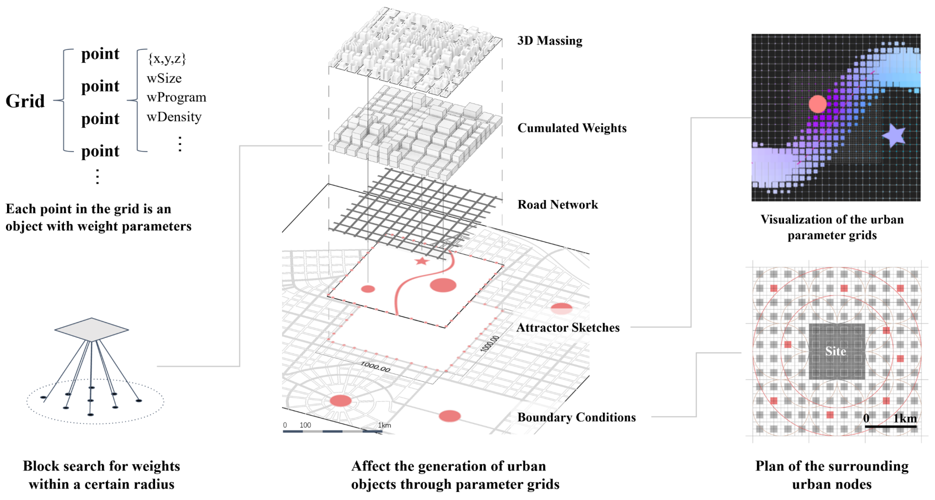
Grid
point
point
point
point
{x,y,z}
wSize
wProgram
wDensity
Each point in the grid is an object with weight parameters
Block search for weights within a certain radius
3D Massing
Cumulated Weights
Road Network
Attractor Sketches
Boundary Conditions
1000.00
1000.00
0
100
1km
Affect the generation of urban objects through parameter grids
Visualization of the urban parameter grids
Site
0
1km
Plan of the surrounding urban nodes

Figure 5. 3D Massing Generation

Step 3 3D Massing Generation: Shown in Figure 5, the 3D massing is further defined by a grid system that incorporates various weighted parameters based on user inputs and surrounding contexts. For instance, a "star" shape is interpreted as representing a "central area". It will have a high influence on the density and program score of the grid parameters around its central point. Then, based on the combined influences of scores, the building base plane will be assigned a weighted height and program parameter that will be represented by the final massing outcomes. Currently, the massing is extruded from the base planes. For further development, the massing and its corresponding parameters can be exported to architectural scale generative design tools, allowing further massing division and detailed architectural design.

Step 4 Visualization and Representation: The prototype's visualization component is powered by Three.js, a widely used WebGL library, to achieve a high-performance real-time rendering of the generated urban environment. The React framework is used to create a control layer that receives user sketches, with the dat.GUI as general parameter input panel. The program featured augmented representation by adjusting material properties based on inherent urban parameters. Furthermore, the user can switch between display modes (perspective and isometric) and perform real-time rendering by additional libraries like three-gpu-pathtracer (Gkjohnson, n.d.). Figure 6 shows the live interactions a user creates with the prototype.

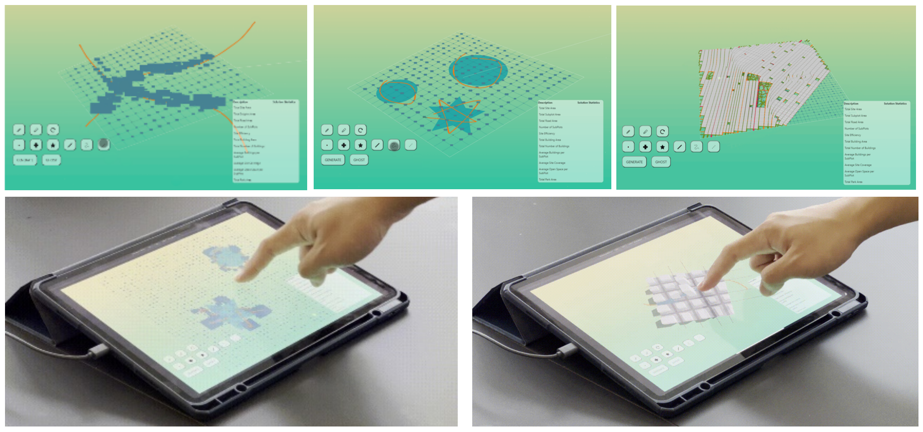
Figure 6. Real-world Use Cases

Step 5 Post-process of Generation Results: The generation results are designed to be exported as compatible formats for architectural scale generative software, ensuring further manipulation and refined design by the next step of workflows. Additionally, the model can be exported as a standalone entity and imported into common modeling software utilized as the foundational model for proposal graphics and further rendering.

4. Discussion and Future Works

4.1. DISCUSSION

Unlike traditional urban design tools or modeling software that often rely on intricate inputs and non-dedicated workflow for urban designers, this tool proposes an intuitive approach by interpreting designers' symbolic languages and sketches into the computer-aided urban design workflow. The tool stresses interactivity, which is fundamental in fitting the actual needs of designers. Currently, with more advanced architecture-level generative design tools coming out, the research aims at filling the gap in urban-scale design tools and allows a seamless transition from urban-scale design to detailed architectural design. The tool can be used in many use cases: First, it can be used by urban planners and designers to quickly conceptualize and manipulate urban layouts, and preview the outcome in a cross-platform, interactive user interface. In academic settings, it can be used as a teaching tool to demonstrate key principles of urban design. The students can experiment with multiple design approaches with an easy learning curve. The tool can also contribute to the participatory design process by allowing non-professionals to contribute to design discussions in a more interactive and intuitive approach.

4.2. FUTURE WORKS

As the prototype is still in its development phase, there are several key areas of enhancement planned for the next phase to make the proposed workflow more robust and adaptable. First, import real-world map data as context, to provide an adaptative urban network based on surrounding environments. Defining more types of tensor fields will make the tool more adaptable in representing multiple typologies of urban networks. Furthermore, by expanding and customizing the symbol language system, we can make the tool adapt to a broad range of users. We foresee opportunities below:

- Seamless cross-scale design tool: Currently, the prototype is a middleware between the user's input and architectural scale design software. This barrier can be broken by establishing dynamic links and automatic sync between tools.
- Collaborative symbol language-aided design: Like real-world design practice, there are usually several people sketching on the same piece of masterplan together. With the collaborative editing feature, multiple designers or community members can discuss and contribute to the design outcome under a shared platform.
- Multi-model symbol-driven modeling: The emergence of large language models provides powerful tools for understanding human language. Following the principle of this paper, we can categorize the designer's lingual symbol vocabularies and integrate them with the workflow, to achieve the connection between language and interactive modeling. Moreover, emerging visualization technologies like augmented and virtual reality enable spatial symbol input and an immersive preview of the results. We believe its corresponding workflow will open opportunities for future urban planning and design with more creativity and possibilities.

Acknowledgments

The authors would like to thank the Digital Blue Foam team for their generous help and guidance during the process of development of the prototype.

References

Alexander, C. (1977). *A pattern language: towns, buildings, construction.* Oxford University Press.

Alonso, L., Zhang, Y. R., Grignard, A., Noyman, A., Sakai, Y., ElKatsha, M., ... Larson, K. (2018). Cityscope: a data-driven interactive simulation tool for urban design. use case volpe. In *Unifying themes in complex systems ix: Proceedings of the ninth international conference on complex systems 9* (pp. 253–261).

Batty, M. (1997). Cellular automata and urban form: A primer. *Journal of the American Planning Association, 63(2),* 266-274. doi: 10.1080/01944369708975918

Beirao, J., Mendes, G., Duarte, J., & Stouffs, R. (2010, 09). Implementing a generative urban design model: Grammar-based design patterns for urban design. doi: 10.52842/conf.ecaade.2010.265

Boim, A., Dortheimer, J., & Sprecher, A. (2022). *A Machine-Learning Approach to Urban Design Interventions in Non-Planned Settlements.* doi: 10.52842/conf.caadria.2022.1.223

Chen, G., Esch, G., Wonka, P., Mu¨ller, P., & Zhang, E. (2008, August). Interactive Procedural Street Modeling. *ACM Trans. Graph., 27.* doi: 10.1145/1399504.1360702

Digital Blue Foam. (2023, December 18). *DBF: AI Generative Design and Spatial Analytics.* Retrieved 2023, December 18, from https://www.digitalbluefoam.com/.

Sidewalk Labs. (2023, December 18). *Delve by Sidewalk Labs - Real Estate Generative Design.* Retrieved 2023, December 18, from https://www.sidewalklabs.com

Gkjohnson (2023, December) *Gkjohnson/three-gpu-pathtracer: Path tracing renderer and utilities for three.js built on top of threemesh-bvh.* Retrieved 2023, December 18, from https://github.com/gkjohnson/three-gpu-pathtracer.

Ishii, H., Underkoffler, J., Chak, D., Piper, B., Ben-Joseph, E., Yeung, L., & Kanji, Z. (2002). Augmented urban planning workbench: overlaying drawings, physical models and digital simulation. In *Proceedings. international symposium on mixed and augmented reality* (pp. 203–211).

Keir. (2023, December). *ProbableTrain/MapGenerator.* Retrieved 2023-12-03, from https://github.com/ProbableTrain/MapGenerator (original-date: 2020-04-01T17:01:11Z)

Kelly, T. (2021). CityEngine: An Introduction to Rule-Based Modeling. In W. Shi, M. F. Goodchild, M. Batty, M.-P. Kwan, & A. Zhang (Eds.), *Urban Informatics* (pp. 637–662). Singapore: Springer. doi: 10.1007/978-981-15-8983-6 35

Koenig, R., Miao, Y., Bu˘s, P., Knecht, K., & Chang, M.-C. (2017). *Interactive Urban Synthesis Computational Methods for fast Prototyping of Urban Design Proposals.*

Lynch, K. (1964). *The image of the city*. MIT press.

Mei, Z., Pan, Y., Cheng, J., & Garcia Del Castillo Lopez, J. L. (2021). Cross-Scale and Density-Driven City Generator—Parametric assistance to designers in prototyping stage. 563–570. https://doi.org/10.52842/conf.ecaade.2021.1.563

Olascoaga, S., & Emilio, C. (2016). *Painting with data: from a computational history of urban models to an alternative urban computing* (Thesis, Massachusetts Institute of Technology). Retrieved 2023-12-03, from https://dspace.mit.edu/handle/1721.1/106680

Olascoaga, S., & Emilio, C. (2021). *Drawing Participation: Histories of Geospatial Computing, Professional Silos, and Computational Potentials for Collaboration in Planning and Design* (Thesis, Massachusetts Institute of Technology). Retrieved 2023-12-03, from https://dspace.mit.edu/ handle/1721.1/140089

KPF Urban Interface(2020)., *Scout.* (n.d.). Retrieved 2023-12-17, from https://scout.build/

Sevtsuk, A. (2021, October). Estimating Pedestrian Flows on Street Networks. *Journal of the American Planning Association*, 87(4), 512–526. doi: 10.1080/01944363.2020.1864758

mrdoob(2023)., *Three.js – JavaScript 3D Library.* (n.d.). https://threejs.org/.

Triantakonstantis, D., & Mountrakis, G. (2012, December). Urban Growth Prediction: A Review of Computational Models and Human Perceptions. *Journal of Geographic Information System*, 4(6), 555–587. doi: 10.4236/jgis.2012.46060

White, E. T. (1975). *Concept sourcebook: a vocabulary of architectural forms*.

Wilson, L., Danforth, J., Davila, C. C., & Harvey, D. (2019). How to generate a thousand masterplans: A framework for computational urban design. In *Proceedings of the symposium on simulation for architecture and urban design.* San Diego, CA, USA: Society for Computer Simulation International

Wilson, A. D., & Li, Y. (2007, October). Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In *Proceedings of the 20th annual ACM symposium on User interface software and technology* (pp. 159–168). Newport Rhode Island USA: ACM. Retrieved 2023-11-26, doi: 10.1145/1294211.1294238

Zhu, W. (1993). *Space, symbol, city: A design theory of city.* China Architecture & Building Press, Beijing.