How to Generate a Thousand Master Plans: A Framework for Computational Urban Design

Luc Wilson[1], **Jason Danforth**[1], **Carlos Cerezo Davila**[1], **and Dee Harvey**[1]

[1]KPF, New York, USA, lwilson@kpf.com

ABSTRACT

The current process for the design of an urban master plan typically involves a team of architects and urban planners that conceive a handful of schemes based on zoning requirements with the help of CAD software. They may intend for the plan to achieve a set of performance goals (economic, environmental, etc.), but quantitative analysis is rarely conducted early and consistently through the design process. This makes it difficult to understand the full range of approaches that are possible on a site, and the relative performance of each scheme. In order to best accommodate rapid urbanization while making cities more sustainable, livable, and equitable, designers must utilize quantitative tools to make informed decisions about their designs. Computational design techniques have been successfully used at the building scale to test numerous designs and quantify their performance, but are challenging to apply at the urban scale due to increased computational expense, difficulty in limiting inputs, and more stakeholders involved in the process. This paper outlines a methodology developed in practice for applying computational design at the urban scale through four steps: 1) Simplified Input Definition 2) Procedural Geometry Generation, 3) Performance Evaluation and 4) Analysis & Communication to generate and test thousands of master planning scenarios.

Author Keywords

Computational Urban Design, Generative Urban Design, Master planning, Urban Planning

1 INTRODUCTION

Issues as diverse as population growth, transportation, and climate change, all present significant challenges for 21st century cities, and require an approach to urban development that is data-driven, iterative, and most importantly, engages the broadest possible audience of stakeholders. Unfortunately, the design tools traditionally available to the architects and urban planners shaping such developments struggle to integrate these needs. In design practice, decisions regarding key urban performance drivers, such as land use, density, and building morphology [16], are often made by refining a small number of schemes, developed through manual iteration, without systematically analyzing the full range of possible designs and their performance implications.

This designer-led time intensive process, can hardly integrate the perspectives of the multitude of involved stakeholders with differing, and often misaligned objectives and expertise. Expert consultants, developers, planning agencies, city councils, community boards, and the general public all bring valid perspectives that must be synthesized into a coherent vision[1].). The authors, through their practice, have worked on 28 master plans over the past 10 years and have experienced these challenges first-hand: key performance metrics must be agreed upon before meaningful design work can commence (a process significantly more complex at the urban-scale, compared with developer-driven, architectural-scale projects); the long timescale of master planning work requires adapting to shifting political priorities[2]); and the final product of the master planning process is not a finished urban form, but rather a series of rules which must be flexible enough to accommodate a range of future development scenarios. Practicing architects and planners require computational tools capable of evaluating performance goals based on the information available at each step of the development of a master plan, and communicating the impacts on those goals of any decisions regarding land use, density and form.

This paper introduces a flexible methodology for Computational Urban Design (CUrbD) as a response to these limitations in current practice, details its application within the Rhino3d CAD environment for the design of a hypothetical district scale development, and discusses three case studies related to urban design and stakeholder engagement. The

[1]The planning process in New York City includes the Community Board, Borough President, City Planning Commission, City Council and Mayor, in addition to the designers, consultants and client [9]

[2]The authors worked on the master plan of Hudson Yards in New York City: started in 1997, initial master plan released in 2001 [24], and last revisions occurring in 2009 [8].

SimAUD 2019 April 07-09 Atlanta, Georgia

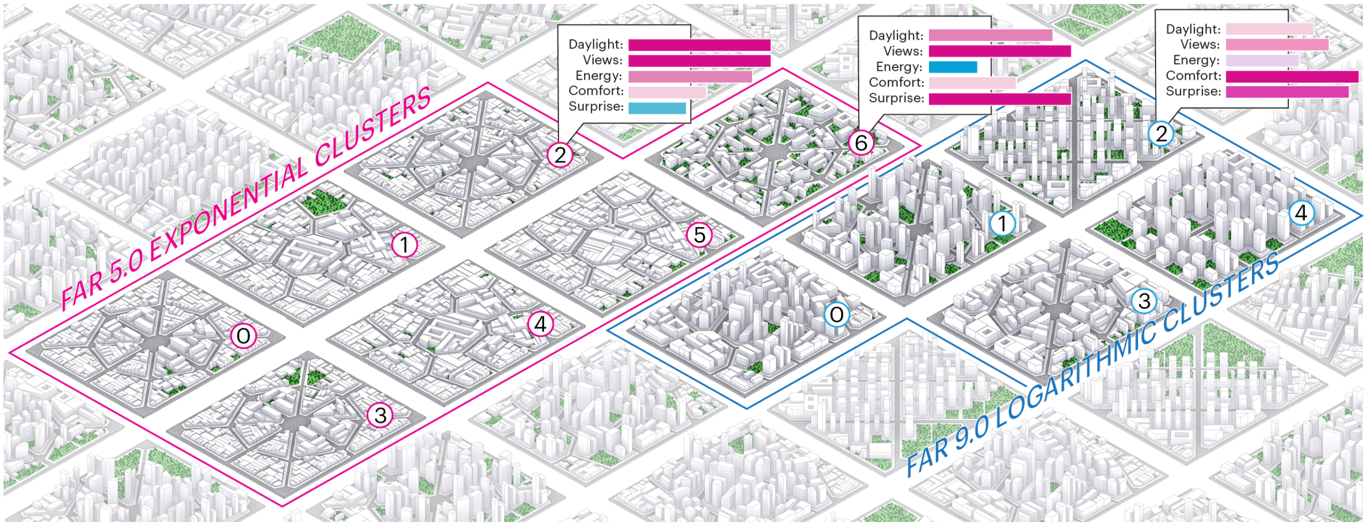

Figure 1. Partial design space of generated master plans, showing representative schemes for each cluster and scores associated with select clusters.

methodology, described in section 2, is structured in four main steps: 1) Input and Design Space Definition, based on generic data formats such as raster images and vector networks, 2) Procedural Geometry Generation of building and block types, 3) Performance Evaluation and 4) Analysis, Communication and Stakeholder Engagement, through visual interfaces and statistical models.

1.1 Related Work

In contrast to traditional approaches to design, where solutions are refined through manual iteration and experience, computational design methods take advantage of parametric CAD tools to explore larger design spaces. They are routinely utilized in architecture and engineering, typically for the optimization of discrete problems, such as building form and faade geometry for structural and environmental performance, and primarily to address the needs of a single stakeholder, the client [4]. More recently, parametric and generative computational models have been proposed as a tool for urban design and planning [12], [22], but their adoption in practice is still very limited [6]. Documented case studies mainly focus on modeling individual aspects of a design [19], rarely tackle the full scale of a master plan [18], and lack the geometric complexity required for their application in practice [3]. To address the problem of generating a sufficiently complex model out of easy to communicate design inputs at a sufficiently large scale, the method here presented proposes to use simple generic raster inputs, rather than full architectural models, to prescribe land use and density, a variation on the Cellular Automata approach to urban form generation by Batty etal [2]. A similar technique is proposed by Stouffs et al, in the design of large project representative of the full master plan scale often found in practice [23], and by Beirao et al [Beirao 2011] as a tool for interacting with a design team. However, the resulting modeling workflow does not offer a way to incorporate the necessary stakeholder engagement.

In addition, most published computational methods are built around urban form optimization, a technique not well suited to accommodate the changing priorities of the many stakeholders involved in the planning process. Nagy etal, present one of the few examples where this approach is applied in practice, presenting a generative design cased study for the planning of a multi block cluster optimized for profitability and solar energy generation [18]. While necessary for understanding technical requirement sin an urban project, this and any similar optimization takes on computational urban design suggest a zero-sum game benefiting a single party [14].

An alternative solution to the optimization of multi stakeholder urban design projects is that proposed in the Urban Simulation Big Data" (URB) method [5]. In it, Cajot etal introduce a multivariate optimization algorithm attempting to balance the goals of all decision makers for the purpose of energy planning, and approximating their likely decisions once the design is complete. However, it still focuses on finding a single best design solution, rather than providing a sufficiently adaptive computational workflow that remains in use throughout an unpredictable, time-intensive master planning process [16]. The methodology introduced in the following sections, takes a different approach, by applying statistical analysis and interactive visualization tools to present stakeholders with families of solutions with distinct pros and cons. This approach, new in an urban application, has been proposed by Mueller for the exploration of structural design solutions as an alternative to pure optimization [15].

2 METHODOLOGY

This section provides an overview of the CUrbD methodology, which can be executed by computational experts separate from the design team, or a computation designer embedded on the design team, with engagement from stakeholders at each step of the process.

1. **Define Inputs & Design Space**
2. **Procedural Geometry Generation**
3. **Performance Evaluation**
4. **Analysis, Communication & Engagement**

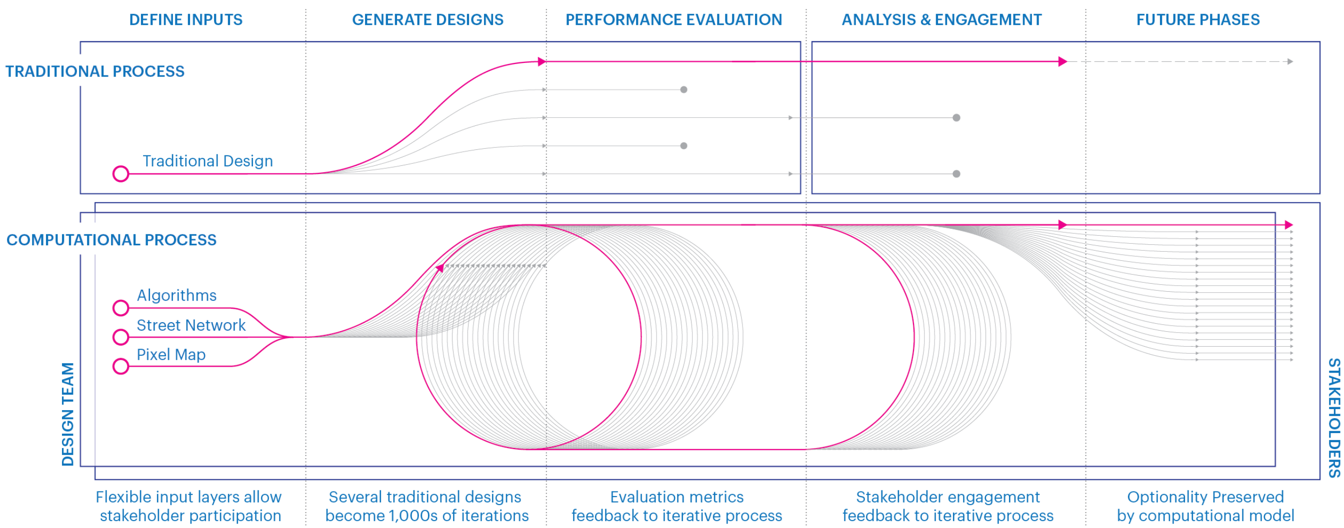

Figure 2. Comparison of traditional master planning process (top) with CUrbD methodology (bottom).

2.1 Inputs & Design Space

In the first step a set of input form related variables are defined to drive variation in a "design space" of thousands of master plan iterations[3]. Initial inputs are grouped in 2 streams that provide all necessary data for design variation while maintaining computational efficiency; one conveying use and density, and another varying street grid, block size, shape and orientation. If instead each building was defined individually by more complex inputs, the design space would quickly get too large to solve reaching millions of options for just one hundred structures[4]. A more complex variation of individual building sis tackled in the procedural modeling step.

The first input, the pixel map (Figure 3) represents a grid applied to the urban site, in which each point stores attributes for use (e.g. "residential") and density (e.g. Floor Area Ratio (FAR) of 4). This map can be generated both computationally or by hand, sketching the combinations of density and uses to be tested, adding flexibility to explore different patterns through the site with minimal drawing work in an accessible format for non-designer stakeholders to contribute in. The second input, the street network (Figure 3) can also be generated analytically or manually, becoming a useful tool during stakeholder meetings where street hierarchy and block sizes can be discussed, and directly loaded into the CUrbD model. As opposed to the latter procedural generation step, the specification of both maps requires input from the designers.

Once both are defined, pixel values are aggregated by block polygon accodring to the streets. Pixel attributes are then assigned to the block with the closest centroid. Small enough pixel dimensions need to be defined depending on the site to guarantee that blocks can contain a mix of uses. Next, density and use values are added together to get the total each use type within the block. Parks and open spaces are the only use treated differently, aggregated to blocks as a percent coverage of their total area (Figure 3).

[3] A design space is the combination of all input variables.

[4] If a master plan has 100 buildings, each with 2 inputs (ex. height and orientation) that would result in 2 to the 150 combinations.

2.2 Procedural Geometry Generation

Procedural generation allows for geometry with the complexity required in practice while maintaining a reasonably sized design space, by defining buildings based on dependant relationships rather than independent inputs. The procedural generation approach outlined in this section is representative of a large family of methods that have been applied in urban modeling in the past [13] [11], but offers a level of complexity appropriate for this application.

In the proposed approach, after the aggregation of the pixel map within the street network, the blocks are split into parcels, by applying a readily available algorithm as part of the Decoding Spaces tool kit [1]. The size and shape of the parcels are generated procedurally based on the density and uses of the block or proximity to other elements of the plan, such as transit or landscape features. For example, low density residential may result in small parcels, while high density residential may result in large parcels. Once the parcels are generated, each one is allocated a portion of the density proportional to it's lot area. They are then populated with building types procedurally generated based on the shape of the parcel, density, and use(Figure 3).

Building typologies for different densities are defined or sketched out in advance with stakeholders to include desired formal characteristics appropriate the project. in the experience of the authors, it is especially important to develop a library of generic low, mid and high density building types that can be modified for each project to avoid generating completely new procedural types. Each iteration resulting from this process is stored as a packaged file that includes pixel map, street network, blocks, parcels, parks, and procedurally generated 3D building geometry. Geometry generation is often much faster than performance evaluation, and separating the two allows the next step to be processed in batches distributed across multiple instance of the modeling environment of choice; in this case Rhino.

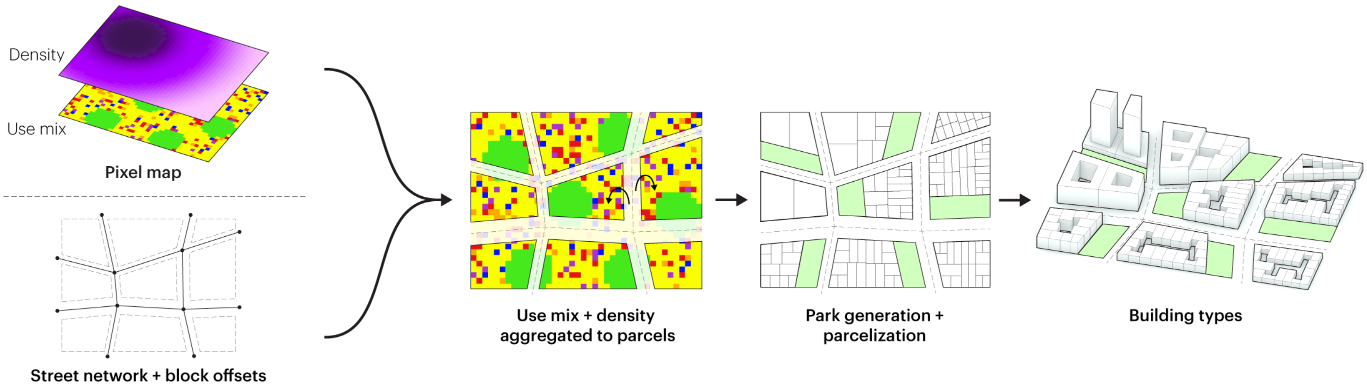

Figure 3. Inputs and procedural generation.

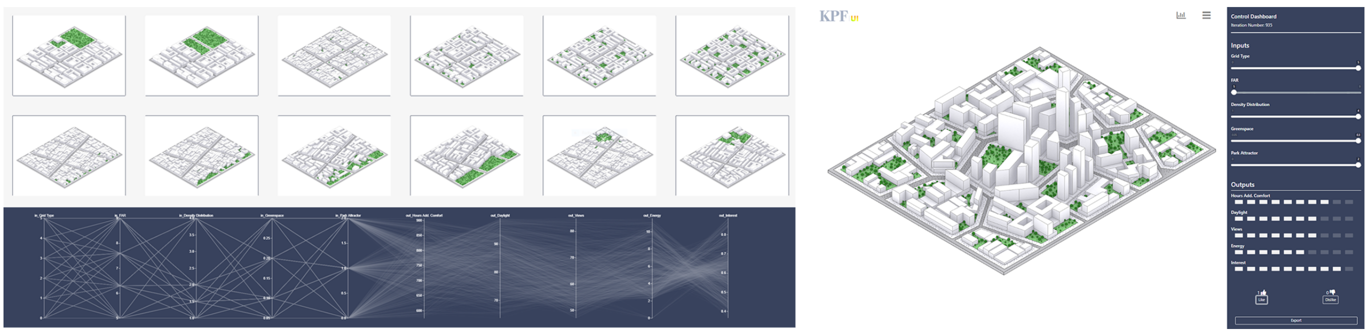

Figure 4. Scout parallel coordinate plots (left) and "explorer" mode (right). URL: kpfui-scout.s3-website-us-east-1.amazonaws.com/SimAUD2019

2.3 Performance Evaluation

Once the design space has been generated and the geometry exported, performance evaluation tools are applied on each iteration to produce a set of analysis metrics. Any type of evaluation (environmental, economic, mobility) can be used based on location specific data (climate, transit, real estate value, etc.) as long as the necessary geometry was generated by the procedural generation model.

Although exploring the optimal analysis tool for each possible urban performance metric is not the purpose of this paper, the authors have found in practice that it is best to select them based on the following criteria: First, tools and metrics should be chosen based on consultation with stakeholders and their specific goals. Second, performance metrics that have direct correlation (e.g. daylight access and views in buildings) should be avoided to reduce computation time while maintaining performance trends. Finally, and the converse to the previous point, metrics that and inversely correlated should be favored for a better understanding of the design (e.g. daylight access VS public space shading ratios). Working with performance tools and metrics at the urban scale produces a unique challenge due to the computation expense of simulation where hundreds of buildings are evaluated. Since CUrbD generates many options for comparison, often relative performance is more important than absolute accuracy as the performance trends will remain the same in either scenario. Computational expense can be addressed by reducing the resolution of the simulation or through the use of proxies, such as the one for urban daylighting established by Dogan, et al. [10].

2.4 Analysis and Interpretation

Results produced by the evaluation tools are dense, multivariate, and challenging to disentangle. However, there are several analysis methods available that can derive actionable insights and drive the design process forward, all while engaging the myriad stakeholders and conflicting agendas typically associated in large urban projects (clients, city governments, designers, and the public).

Filtering & Visual Exploration. The most rudimentary approach to analysis is to export results into a spreadsheet and sort the metrics for minima or maxima. This allows the user to filter out low performing options, but is insufficient for multivariate trends, and ineffective for graphic communication. Visual Interfaces, such as web-based data visualization tools like Core Studio's Thread [7] and Scout, which was developed by the authors, allows users to explore, sort, and filter the design space of iterations based on their relative performance. Scout features two interfaces that anticipate different levels of user sophistication. The first has a side menu containing sliders that allow the user to set the input values and view each iteration one by one. Metrics are displayed below the inputs so the user can understand the performance of one option at a time, hiding the complexity of the design space and making exploration more accessible. The second interface introduces a parallel coordinates plot that provides dynamic exploration, allowing users to set bounds for an individual metric, and then revealing the remaining iterations and their scores across the other criteria (Figure 4).

These tools quickly surface strong trends in the results, and engage a broad audience of non-experts in exploring the data. At the same time, the emphasis on accessibility and intuitive user experience does limit the sophistication of the analysis, particularly regarding weak, or inconsistent trends that hold for one subset of the data but dissipate elsewhere.

Correlation & High Performance Design Trends. Of the many tools available within descriptive statistics, Correlation Matrices produce a visual summary of the relationship between every parameter of the model, giving designers immediate insights into the association of each input (form) to each output (performance,) as well as each output to each output (the association of inputs with inputs is predetermined by the design of the model). Rather than using a Visual Interface to filter through every value for a given input and tracking the results for each output metric, the viewer can simply scan across the relevant row in the matrix and see the numerical correlations. This approach can be used to establish a framework of performance-based design guidelines that can then be graphically communicated to the team, but is worse suited for engage non expert stakeholders (Figure 5).

Clustering & Establishing Design Schemes. Moving one step beyond performance-based guidelines, Unsupervised Learning (a subset of machine learning methods), allows the data-set to speak for itself [25], auto-generating trends based on myriad relationships in the results, both strong and weak. The most common form of unsupervised learning is clustering, which groups iterations together into coherent sets of relationships (i.e. a combination of particular inputs that lead to a consistent sets of results). Designers can then use each cluster as a starting point to direct the early stages of a project, or a family of similar urban solutions which can be easily communicated to stakeholders.

The Scikit-Learn implementation of the k-Means algorithm is both user-friendly and broadly applicable to computational design, but it is also fairly generic, and several steps should be taken to ensure meaningful results. First, breaking the data into subsets will generate more specific trends. These subsets can be based on whichever criteria is most relevant to the project, and since density (FAR) is the central driver of building typology in the CUrbD model (and a prime concern for cities and developers alike), it is used here to separate the data. Second, it is important to recognize that k-Means clustering will include every iteration in one of the clusters, so filtering out low-performing options will increase the clarity of relevant trends. Given that overall performance should be relative to all the evaluation metrics, this filtering can be accomplished one of two ways: either a "most-of-the-best" approach that applies an exponential function to identify candidates with the highest possible scores across the greatest number of metrics, or a "least-of-the-worst" approach that applies a logarithmic function to avoid candidates with low scores on any single metric. Lastly, k-Means does not determine the optimal number of clusters automatically, but this can be found by testing a range of options and solving for the best balance of the Silhouette parameter, which indicates the degree of separation between each cluster, and Distortion parameter, which measures the distance between each observation and the centroid of the cluster.

3 DEMONSTRATION

In order to demonstrate the CUrbD methodology, we applied it to a hypothetical, rectangular site with the climate profile of Toronto, Canada, and produced a design space of 1,152 iterations. For the procedural generation of these iterations we used Rhinoceros, a computer-aided drafting program, and Grasshopper, which is a graphical scripting plugin for Rhinoceros [21] [20]. We augmented Grasshopper with python scripts to handle some of the more complex geometry and file management. The parcelization algorithm was supplied by the plugin Decoding Spaces [1].

Define Inputs & Design Space. In order to generate variation across the master plans, we varied 5 of the inputs (street network, density, density distribution, park space %, and park attractor), and applied a "brute force" method of cross referencing the variables in Grasshopper to ensure every possible permutation of inputs was tested. This is a markedly different approach than optimizing for top performing designs (i.e. using a genetic algorithm or similar method), and while it has a significantly higher compute time, it creates the potential for a far more engaging exploration of the results, particularly for a general audience whose design preferences are not known beforehand.

Some input values, like "street network" and "park attractor" are simply an index that tells the script to import manually drawn geometry. For instance, there were 6 options for the street network. Choosing one of the indices from the range 0-5 would determine which of these street networks would be imported. The same was done for the park attractor. The park attractors were simply geometry that the parks would cluster around. We used 3 park attractors: the first were dozens of points distributed evenly across the site to simulate a distributed park scheme, the second was a single point that would generate a centralized park, the last was a line along one edge of the side simulating a linear park. The rest of the inputs represented specific values that were communicated to the model. "Density" was the total FAR for the site. This was communicated to the pixel map by providing an even FAR across the entire site. "Density distribution" redistributed this density so that it peaked in the center of the site. The value for this input represented the ratio of the least dense pixel, to the most dense pixel in the center of the site. The park space % represented the percent of pixels that would be designated as park space.

Procedural Geometry Generation. We then ran the procedural generation algorithm on each design iteration to produce a design space. In order to demonstrate how buildings could populate the site, we used a diversity of building types that we distributed based on 3 density categories. High density blocks were populated with either a tower/podium type, or a simple lot-line extrusion if the lot was too small. Medium density parcels were given either a courtyard building, or a simple extrusion if the lot was too small. Small parcels all received a simple extrusion with a rear setback. While these are the particular rules and building types we implemented for this example, the metholdogy is exible and can be implemented with any rule based building type and distribution logic. Once the procedural geometry generation was complete, we saved each iteration as CSV for the pixel map, json for the street network, and a 3dm file for blocks, parks, parcels, and buildings.

Figure 5. Correlation Matrix showing relationship between all inputs and evaluation metrics.

Performance Evaluation. For this study, we produced a score for each of the following: views, daylight, comfort, energy generation, and visual interest. For facade based evaluation tools, like views and daylight, we created a grid of sensor points on each facade. ”Views” refers to the unobstructed percentage of human field of vision, while ”Daylight” measures the vertical sky angle for each facade point. In the analysis of public space, ”Visual interest” evaluates the variation of density between adjacent study points and ”Comfort” evaluates the Universal Thermal Climate Index (UTCI) index. Last, ”Energy generation”, which measures solar PV potential, generates an analysis grid only on building roofs. We then export the scores from each metric, along with their associated inputs, to a CSV of results that can be used for analysis.

Analysis. To analyze the results of the study, we followed the steps outlined above in the Methodology section. First uploading results into a visual interface for initial exploration and testing. Second producing a correlation matrix to understand the fundamental relationships between design parameters and performance metrics, and to either confirm macro assumptions, or identify areas of interest within the data; Next creating subfamilies of design options based on the density values, and generating summary for each (both the most-of-the-best exponential score, and the least-of-the-worst logarithmic score) to filter out low-performing iterations. Finally, k-Means clustering was implemented to group observations together into the most relevant trends allowing for categorization of high-performing and distinct design schemes. Each one of these analysis tools was presented and discuss to the design teams working with the authors.

The results for the 5.0 FAR clusters (Figure 1) show that while there were seven distinct clusters, there were only three main trends: Clusters 1, 4, and 5 all had a low Density Distribution and the voronoi-based Grid 5, while clusters 0, 2, and 3 had a higher Density Distribution and the radial Gird 4. All six of these clusters had the lowest possible amount of parks (5%), and all six scored similarly across the evaluation metrics. This strongly suggested that the best performing low-density iterations would privilege Daylighting, Unobstructed Views, and Energy Generation, with above average scores for Outdoor Comfort, and low scores for Visual Interest.

In terms of inputs, we could be confident that minimal parks, and either 1.0 Density Distribution / Grid 5, or 2.0 Density Distribution / Grid 4 would return the best results. Contrariwise, Cluster 6 provided an interesting alternative strategy to privilege Visual Interest and Views at the cost of Energy Efficiency, but it should be noted that it contained only a single observation out of the twenty represented across all seven clusters. This made it a much less robust trend, and therefore far less likely to preserve its performance as the final design inevitably deviates from the simplifications of the iterative model. Taken together, this data-driven analysis, combined with the 3-D geometry and visualizations produced by the model, provided clear design direction for any low-density scheme on this site, while preserving optionality and setting expectations for performance.

To fully explore the trends, the k-means clustering process was repeated for each density (FAR 5.0, 6.3, 7.6, and 9.0) and for each scoring approach (i.e. the Logarithmic and Exponential methods). While it is beyond the scope of this paper to report on the full conclusions of this analysis, the results for the Logarithmic scoring of FAR 9.0 are included for comparison, and it is worth noting both the extreme variation in the geometry of each cluster, as well as the more balanced results (Figure 1.)

4 APPLICATION IN PRACTICE

This section illustrates the CUrbD methodology through its application in three real projects, and outlines best practices for successful implementation.

A new district in Hangzhou. We used CUrbD to create a design tool for a 620 acre master plan in Hangzhou, China to create a new mixed-use district. Here CUrbD was used to address a discrete challenge in the planning process. Federal regulations require a minimum duration of direct sun on the winter solstice for residential units (two hours in Hangzhou). Typically, this regulation results in a modernist tower-in-the-park building type, making it difficult for the design team to achieve their intent to create smaller blocks, continuous street walls, and narrower streets. To address this challenge, we defined inputs as a range of block size, street width, gross floor area, and street wall height resulting in a design space of 7,400 options (A pixel map was not used since it was a single use project). From those inputs we applied a procedurally generated courtyard-with-towers block type, which was then evaluated for compliance with the direct sunlight regulations for residential buildings in China. We uploaded the results into Scout and provided the app as a tool for the design team.

Figure 6. Interactive interface for Sidewalk Toronto

They used the parallel coordinates chart to filter for the desired inputs, such as street width and target GFA, and picked from the complying options. It allowed them to find solutions that achieved the kind of urban character that they desired, while meeting the regulations, without defaulting to a typical tower-in-the-park urban design scheme.

Stakeholder engagement for Sidewalk Toronto. Working with Sidewalk Labs, we developed a CUrbD model to assist with the master planning of the Sidewalk Toronto project (URL: https://sidewalktoronto.ca/). As part of a public-facing exhibition at their Toronto workspace we ran the model for an abstract site with inputs that included a representative sampling of options under consideration for the waterfront development, as well as more experimental edges cases that featured lower and higher densities, abstract street grids, and ambitiously large green spaces. (The model was very similar to the example in the demonstration section and with the same performance evaluation criteria.) The results of this model where used for a physical interface that allowed the pubic to engage with the CUrbD process (Figure 6). Visitors explored combinations of density, public space and street grids by toggling wooden knobs to change design inputs. This allowed users to create the type of neighborhood they wanted and to then understand how those design decisions impact the functioning of a complex system like a city, encouraging design and introspection in equal measure. For example, one participant started with the lowest population and the most green space (she wanted a backyard of her own), but quickly realized that this led to low scores for outdoor comfort and energy efficiency (two things she valued). By making a few quick adjustments she found an option that performed well for those two priorities. Looking ahead to future implementations, this sort of user engagement could also be recorded, aggregating participant feedback into implicit, qualitative metrics which could, in turn be used to drive further generation of additional design options. [17]

Technology Campus in Southern China. Lastly, we applied CUrbD in the design of a 30 million sq ft master plan (mostly RD and residential with some retail and event space) in a hot, humid city in southern China (the actual location and client are confidential). The application of the methodology happened in parallel with the design team. Ideally, the methodology is used prior to the design team starting on a project, which is often not possible. This example will outline approaches for application in the often not ideal circumstances that occur in practice.

To compliment the design as it developed in parallel to our work, we focused the analysis of the CUrbD process on recommendations specific what was still flexible in the design scheme, such as changing massing orientation and program distribution in order to reduce solar radiation and decrease average trip duration. To do this we established a combination of inputs that were computational derived and manually drawn by the designers. Next, we developed procedural versions of the building types being developed by the design team. This allowed us to tailor design guidelines (using the correlation approach in section 2.4) to the design issues that could still changed within in the design. When they integrated our guidelines into their scheme, they resulted in increasing outdoor comfort by 33.7%, decreasing average trip duration 24.7%, and decrease solar radiation on buildings by 15.2 % when compared to initial design.

As illustrated through application in practice, effective communication of the results of a CUrbD can be difficult, but is crucial for it to have meaningful impact in the master planning process.

5 DISCUSSION & NEXT STEPS

5.1 Challenges

A challenge of this methodology that requires further work is the relationship between form and performance. At the building scale, if you change height, orientation, or location, the link to the resulting performance is clear. At the urban scale, performance is being analyzed across a heterogeneous urban fabric. This means different parts of the master plan can perform differently. When you distill the analysis of the master plan to a single metric, most of this variation is lost. For instance, in the same master plan there may be one group of short buildings which score poorly for the view score, whereas a group of tall, widely spaced buildings score extremely well. An average of these view scores would not reflect the variation of the site or the equity of the score. Further development of analysis tools will focus on addressing the spatial distribution of the performance evaluation.

Because the CUrbD process is composed of algorithms, it would be a mistake to think that its unbiased. The range of values supplied for inputs could exclude certain possibilities that might be desirable to some stakeholders. One solution to limit bias is to provide a much larger range of options in terms of the inputs and logic upon which the model is built. Another solution is to solicit specific inputs from all stakeholder since this methodology allows for manually generated inputs. The potential for bias also illustrates the need for design and judgment in the CUrbD process and the active engagement in with stakeholders so that, while not every option is explored, the critical ones are represented.

6 CONCLUSION

While computational urban design shows much promise for providing an iterative, quantitative approach to master planning, its place within the master planning process remains in question. While we've shown how the CUrbD process can generate useful insights in real projects, how these insights influence what actually gets built is unknown. These insights must be utilized within the complex, multi-stakeholder environment of both the design process, and the implementation of the master plan over the long term. As a result, computational urban design, at least initially, needs to work in coordination with the traditional master planning process. However, with the increasing challenges of population grown, transportation, and climate change, master planning must take advantage of iterative and quantitative approaches to urban design. A process such as CUrbD provides an opportunity to navigate the myriad of seemingly contradictory constraints and stakeholder interests of a master planning project.

REFERENCES

1. Abdulmawla, A., Bielik, M., Buš, P., Mei-Chih, C., Dennemark, M., Fuchkina, E., Miao, Y., Knecht, K., König, R., and Schneider, S. Decoding Spaces, 2017.
2. Batty, M. Cellular automata and urban form: A primer. *Journal of the American Planning Association 63* (06 1997), 266–274.
3. Beiro, J., Mendes, G., Duarte, J., and Stouffs, R. Implementing a generative urban design model: Grammar-based design patterns for urban design. *28th eCAADe, pp.265-274* (09 2010).
4. Brown, N. C., and Mueller, C. T. Design for structural and energy performance of long span buildings using geometric multi-objective optimization. *Energy and Buildings 127* (2016), 748–761.
5. Cajot, S., Schüler, N., Peter, M., Koch, A., and Maréchal, F. Interactive optimization for the planning of urban systems. *Energy Procedia 122* (2017), 445–450.
6. Calskan, O. Parametric design in urbanism: A critical reflection. *Planning Practice & Research 32*, 4 (2017), 417–443.
7. Core Studio. Thread. `http://core.thorntontomasetti.com/thread/`, 2019.
8. Department of City Planning, City of New York. Hudson Yards. `https://www1.nyc.gov/assets/planning/download/pdf/about/press-releases/pr121201.pdf`, Jan 2009.
9. Department of City Planning, City of New York. Uniform Land Use Review Procedure (ULURP). `https://www1.nyc.gov/site/planning/applicants/applicant-portal/step5-ulurp-process.page`, Accessed: February 2018.
10. Dogan, T., Reinhart, C., and Michalatos, P. Urban daylight simulation calculating the daylit area of urban designs. *SimBuild 2012* (01 2012), 613 – 620.
11. Esri. Esri City Engine.
12. Fusero, P., Massimiano, L., Tedeschi, A., and Sarah, L. Parametric urbanism: A new frontier for smart cities. *Planum. The Journal of Urbanism 2* (September 2013), 1 – 13.
13. Halatsch, J., Kunze, A., and Schmitt, G. *Using Shape Grammars for Master Planning*. 01 2008, 655–673.
14. Lindblom, C. E. The Science of "Muddling Through". *Public Administration Review 19*, 2 (1959), 79–88.
15. Mueller, C. T., and Ochsendorf, J. A. Combining structural performance and designer preferences in evolutionary design space exploration. *Automation in Construction 52* (2015), 70 – 82.
16. Mueller, J., Lu, H., Chirkin, A., Klein, B., and Schmitt, G. Citizen Design Science: A strategy for crowd-creative urban design. *Cities 72*, August 2017 (2018), 181–188.
17. Nagakura, T. *Acadia 2017*. Acadia Publishing Company, 2017.
18. Nagy, D., Villaggi, L., and Benjamin, D. Generative urban design: Integrating financial and energy goals for automated neighborhood layout. *SimAUD 2018* (06 2018), 25.
19. Rakha, T., and Reinhart, C. Generative urban modeling : A design work flow for walkability-optimized cities. *SimBuild 2012* (2012).
20. Robert McNeel & Associates. Grasshopper, 2014.
21. Robert McNeel & Associates. Rhinoceros 5.0, 2018.
22. Schnabel, M. A., Zhang, Y., and Aydin, S. Using parametric modelling in form-based code design for high-dense cities. *Procedia Engineering 180* (2017), 1379 – 1387. iHBE Conference 2016.
23. Stouffs, R., and Janssen, P. *A Rule-Based Generative Analysis Approach for Urban Planning*. 10 2016, 125–136.
24. The Office of the Mayor, City of New York. Mayor Giuliani Unveils Plan For Development Of Far West Side In Manhattan. `https://www1.nyc.gov/assets/planning/download/pdf/about/press-releases/pr121201.pdf`, feb 2001.
25. VanderPlas, J. *Python Data Science Handbook*, 1st ed. O'Reilly Media, Inc., 2016.