SKETCH TO BUILD:
AN INTUITIVE DESIGN PLATFORM FOR SUSTAINABLE HOUSING COMPLEXES

ZhongMing Peter Zhang

New Jersey Institute of Technology
Newark, New Jersey, USA
zpz2@njit.edu

Taro Narahara

New Jersey Institute of Technology
Newark, New Jersey, USA
narahara@njit.edu

ABSTRACT

Today, there is a growing demand for housing complexes due to rapid urbanization in major metropolitan areas. While architects must meet new sustainability standards, they are also expected to demonstrate creative solutions for humanizing mass housing for the well-being of residents. This paper proposes an intuitive platform for users to visually study possible housing complex designs and their potential performance in energy use intensity (EUI), environmental, and some financial criteria based on preliminary sketches drawn by users. Before users start sketching, our program auto-generates basic layouts with performance results. With this knowledge, users will be able to visually grasp intrinsic relationships between built forms and performance characteristics and reflect on their new design. Our goal is to provide a platform that enables designers to effectively incorporate qualitative contributions from early exploratory stages into advanced design stages, allowing architects to focus on more creative solutions.

Keywords: housing complexes, built forms, sustainable performance.

1 INTRODUCTION

Today, rapid urbanization in major metropolitan cities has resulted in an increase in global demand for housing complexes. Simultaneously, new standards for sustainable development require us to reconsider the forms and functions of architecture that meet emerging needs for future cities. Architects are constantly under pressure to meet financial and practical criteria such as base building efficiency (BBE), design compliance, and energy use intensity (EUI). Due to the unprecedented increase in demand, efficiency and speed appear to be prioritized in the professional practice of a commonplace housing complex project today. While there are some emerging tools to auto-generate building schemes with environmental and financial performance measures to aid professionals' productivity, the experience with these tools is not as intuitive as the natural experience of sketching on paper. The development of design tools that can facilitate the connection between early stages of visual thinking activity and more matured stages of development and refinement has been highly sought after. Such tools have the potential to improve the quality of our future housing complexes.

In this paper, we propose a platform that allows users to quickly see how their preliminary built forms for housing complexes could perform based on criteria such as EUI, environmental qualities, and some financial aspects. Before users begin sketching their initial layouts, our proposed platform automatically generates typical basic layout examples based on user-defined site conditions and shows their estimated performances in nine criteria visually represented through radar charts, allowing users to develop better intuitions for relationships between built forms and their performances. Our goal is to provide a platform that allows designers to more effectively incorporate qualitative contributions at early exploratory stages

ANNSIM '22, July 18-20, 2022, San Diego, CA, USA; ©2022 Society for Modeling & Simulation International (SCS)

into advanced design stages for a wide range of users, from novice designers to practitioners. We hope that our proposed platform will increase designers' productivity and creativity so that they can focus on more pressing issues, such as humanizing commercial mass housing for residents' well-being and establishing a better synergy with our built environment.

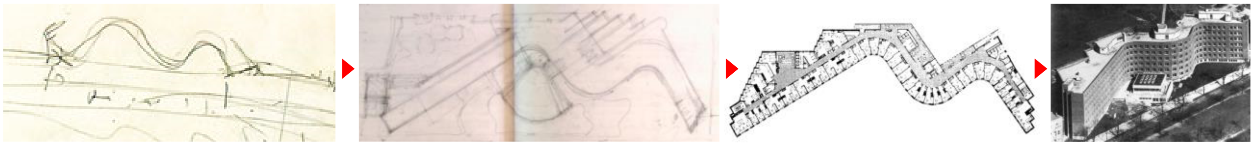

Figure 1: Initial sketches influence the final built form. MIT Baker House, Cambridge, MA. by Alvar Aalto. A sketch and perspective (left: 1947), a sketch (middle; 1948), and a floor plan and a photo (right; 1949).

2 RELATED WORKS

Architects have traditionally worked and interacted with sketches. Many architects and researchers, including Le Corbusier, Louis I. Kahn, and Alvaro Aalto, have emphasized the importance of freehand sketches and diagrams in the design process (Kahn 1931, Do et al. 2001, and Aalto 1947 in Figure 1). The need for a smooth transition from an early exploratory stage of creative activities on paper to a refinement stage using computers has been keenly sought in the visual design task domain beyond architecture. The creation of user interfaces that can aid in the early stages of visual thinking activity on computational systems has been a focus of research in the computer graphics community (Olsen et al. 2009).

In architecture, component-based graphic programming and scripting platforms such as *Grasshopper* and *Dynamo* (2022), embedded within essential CAD and BIM tools, including *AutoCAD*, *Revit*, and *Rhinoceros*, have significantly increased architects' involvement in computational design. However, for average architects to learn technical materials and build an operational workflow, these tools necessitate a significant upfront investment in time. To expand access to such advanced features beyond a small group of architecture professionals, such as developers, urban planners, and even accountants, some emerging tools have focused on ease of use in their user interfaces. For example, *TestFit* (2022) is one of the most widely used commercially available tools for conducting a proforma analysis for housing complexes by real estate professionals. There are also emerging design tools, such as *Spacemaker* (2022) and *Delve* (2021), that are enhanced by data-driven approaches and AI algorithms for auto-generation of design patterns with feasibility analysis.

There has also been research into built forms and the performance criteria for housing complexes. Steadman (2014) used empirical data to investigate relationships between daylight and built forms, such as the volume-to-wall-area ratio in day-lit blocks. Several studies investigated the relationships between built forms and energy consumption (Depecker 2001), occupant visibility and perceptions (Fisher-Gewirtzman and Polak 2019, Schwartz 2021), and density and the quality of living space (Chan et al. 2002). Narahara and Yamasaki (2019) investigated the relationships between forms in residential architectural floor plan layouts and subjective evaluation scores based on functionality and comfort. Some works use multi-objective optimizations based on performance criteria to find variations in forms (Narahara and Terzidis 2006; Narahara 2010; Gerber 2012; Christodoulou 2018; Titulaer 2019).

While some emerging tools enable efficient production and analysis without the need for architects to manually draft buildings from scratch, they still require architects to master specific skills and require some time and effort to retrieve the results. Furthermore, the experience of using these tools is far from the natural experience of sketching on paper with a pencil. While multi-objective optimizations can search forms based on quantifiable criteria, artists require a system that can incorporate their intuitive qualitative contributions from early exploratory design sketches to more advanced stages. Anyone wishing to design a housing complex may find it useful to have a tool that can quickly suggest the potential performance of buildings from a few simple strokes of lines with parameters representing layout options. We propose such a platform for designers in this paper.

3 METHODS

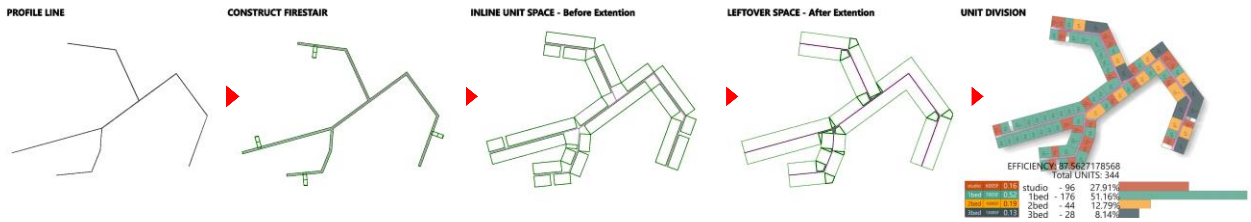

Figure 2: Sequential diagrams illustrating the steps to generate a floor plan using the proposed method in Section 3.1, from a user-drawn sketch (left) to the final floor plan layout (right).

In this section, we introduce our program to generate housing complex layouts. Users of our program are expected to go through the following three steps iteratively to refine their designs. To begin, our program prompts the user to enter the design constraints for apartment units and site conditions described in Section 3.1.1. Second, our program employs the steps in Section 3.1 to automatically generate up to 52 basic building layouts introduced in Section 3.1.2 and display results based on nine performance criteria described in Section 3.2. Third, using the methods described in Sections 3.1 and 3.2, the user can begin a freehand sketch to generate original building layouts. *Grasshopper* (GH) in *Rhinoceros* 7 on a Windows PC with an Intel Core i9 clocked at 4.8GHz was used to implement our proposed program. We wrote some custom code in *Grasshopper* using Python components while using the software's graphics engine, drafting interfaces, components, and plugins for analysis. In the following sections, we will explain our methods.

3.1 Floor Plan Generation

3.1.1 Overview of Design Constraints

Before beginning a freehand sketch to design a building layout, a user must first provide a site condition, which includes property lines and neighboring building volumes that may influence the environment. In addition, the user must enter several parameters that are typically determined earlier in the schematic design phase in professional practice. The parameters include the total number of units, the depth of apartment units, and percentages representing preferred proportions of unit types, all of which are critical values influencing the overall performance of a housing complex. We chose three standard unit types, including studio, 1-bedroom, and 2-bedroom apartments, and the unit areas are adjustable by the user. (The default sizes are 600 square feet (sf), 700sf, and 1000sf based on the authors' region's standards.) Minimum unit length constraints are also required because fire codes require at least one window in each bedroom and living room. The total number of units and the percentage of different unit types are suggested target values for our proposed layout generation process. Any deviations from the user-defined goals are quantitatively and visually represented by our radar charts, which are described in Section 3.2.7.

3.1.2 Sketching a layout

In the second step, our proposed program creates a floor plan layout for a multi-residential building using a user's freehand sketch as the geometry's center lines. Our program instructs the user to begin a freehand sketch by drawing a polyline composed of straight-line segments inside the site's setback lines for buildable areas. Our interface employs a drawing feature based on polygonal segments, allowing the CAD software to accept mouse input. The user creates line segments by utilizing CAD software features such as point snapping. The user can also add branching conditions to polylines and freely generate a large number of building layout variations. Users can update and revise sketches iteratively until they are satisfied.

3.1.3 Layout Variations

Before a user begins creating custom drawings, our program can generate up to 26 pre-defined layouts based on commonly used geometries for multi-family buildings. The pre-defined layouts include I (linear bar), O (courtyard), X (cruciform), ∃E (king), A, C, E, H, and L-shaped geometries, each with several variations for different orientations, positions, depths, heights (low-rise or tall towers), and proportions. These pre-defined layouts are generated automatically based on the property lines of the site using our rule-based procedures written in GH Python. First, preliminary centerline geometries are generated and automatically developed into built forms using the same steps described in the subsections below, but with user-defined sketches. They are then evaluated automatically based on performance criteria (see Section 3.2), with radar charts displaying the results. While some layouts, such as towers, may appear unrealistic under certain design compliance conditions, such as zoning, we decided to include a diverse range of forms for the sake of informing all levels of users about the primary relationships between forms and performances. At the schematic level, the user can intuitively grasp the relationships between built forms and their performances based on various criteria and quickly incorporate their learning into new sketches. A user can choose which examples to analyze because there is a tradeoff between the number of examples and the total time for analysis.

3.1.4 Footprint Generation

Our program converts user-inputted centerline sketches into several space types, including corridors and spaces reserved for vertical circulation. First, our program generates a single floor plan based on a double-loaded corridor using the user's centerline and unit depth. As of today, a double-loaded corridor is the most commonly used floor plan layout in the US real estate market, and we used it as the project's starting point for the time being. According to Steadman, a sufficient day-lit plan depth is 27 feet for the units (2014). However, depths of up to 30 feet are commonly found in real-estate-housing projects directed by developers in the United States in order to achieve higher base building efficiency. We created layouts with 25' and 30' widths for each of our pre-defined 26 patterns (for a total of 52), with 30' being more effective for rentable areas and 25' being more generous for daylight. Six feet is the default corridor width in our system, but it is still adjustable.

To comply with emergency evacuation rules, we created a rule-based code based on the two conditions listed below to place fire stairs inside the open single floor plan. 1) Stairs should be placed no more than 50 feet from a dead-end corridor. 2) Stairs should be separated by at least 250 feet. Then, inline spaces on both sides of corridors are defined, which will later be filled with units. Outside corners with angles greater than 90 degrees and inside corners with angles less than 90 degrees are extended to create more space for units. To improve base building efficiency, our program searches for all non-rectangular residual spaces, such as triangles and quadrilaterals, and incorporates them into adjacent inline unit spaces.

3.1.5 Unit Arrangement Optimization

To divide the inline unit spaces while keeping the percentages of unit types as close to user inputs as possible, we resort to a stochastic optimization method, simulated annealing (Kirkpatrick 1984) with a Metropolis-Hastings state-search step (Metropolis et al. 1953; Hastings 1970) to search for a good approximation to the global optimum. As we use the constant depth for all units, determining segments' lengths by dividing the areas of each inline unit space by this constant depth, we reduce this problem into a more straightforward one-dimensional combinatorial optimization similar to the bin-packing problem (Martello et al. 1990): Given a set of three unit types, each with a width, determine the number of each unit type to include in a collection (or collections of segments) so that the total length is less than or equal to a given limit(s) and the percentages of unit types are as close to user inputs as possible. To approach this problem, we use simulated annealing to find a pair of numbers (counts) for three unit types that can minimize the sum of left-over lengths from each inline unit segment (the first term of (1)) and deviations from the user-defined percentages for numbers of unit types (the second term of (1)). The cost (objective) function is defined as

$$C(x) = \sum_j^m \left(L_j - \sum_i^n a_{ij} A_i\right) + \tau \sum_i^n \left|P_i - \frac{\sum_j^m aij}{T}\right| \quad (1)$$

where $x = \{aij \mid i = 1, \dots, n,\ j = 1, \dots, m\}$ represents a given layout comprising *aij* as a number of the unit type *i* (n = 3: total types) in the *j*-th segment (m = total segments), *Lj* is a length of a *j*-th segment, and *Ai* is a length of a unit type *i*, where $L_j - \sum_i^n a_{ij} A_i \geqq 0$ is kept. *Pi* is a user-defined target percentage for the number of the unit type *i*, T $(= \sum_i^n \sum_j^m a_{ij})$ represents the total number of units, and τ is a constant indicating the relative importance of the term. We run an optimization 500 times and sort the retrieved results by cost.

We provide options for users to select from lists of top-ranked solutions based on the cost function, cost based only on the first term (less space wasted), or cost based only on the second term, as our goal is to show reasonable schematic layouts as good approximations to the optimum (closer to the user-defined percentages for numbers of unit types). The optimized solution for unit counts for 1-D lengths of each segment is translated back into the original profile geometry of each inline segment, while correctly adjusting the 1-D lengths in areas with irregular (non-rectangular) end conditions of a segment. Any residual length in each segment is added to any unit that has less than 50% of a potential window opening area on any side if it exists; otherwise, the segment is divided evenly based on the resulting numbers of three unit types (some units might exceed but are not less than the three user-defined minimum unit sizes).

The available number of units on this single-level floor layout is arrayed and stacked up to the number of floors that comes closest to meeting the user-defined total number of units. Because the purpose of our program at this stage is to represent the initial schematic massing design, we believe that all floors should be identical to maximize construction efficiency.

3.2 Evaluation Criteria

Our program visually displays analysis results from up to 52 sample layouts with radar charts based on the user-defined site conditions and requirements before a user begins sketching. Thus, the analysis results from primary built forms can be intuitively reflected upon by the user to generate subsequent new layouts. All layouts were evaluated using the nine criteria outlined below.

3.2.1 Base Building Efficiency (BBE)

A well-established metric used in real estate industry is the *base building efficiency* (BBE) (Pena and Parshall 2012), which is expressed as

$$Base\ Building\ Efficiency = UsableArea/BuildingGrossArea$$

BBE denotes the square footage that the client can rent to the tenant, excluding shared areas such as the corridor and lobby, in relation to all areas that the client paid architects and contractors to build. This can directly translate to an increase in Net operating income in a more detailed proforma analysis. This figure typically ranges between 75% and 85%. (Pena and Parshall 2012). The higher the BBE score, the more building areas that can potentially generate profits for owners by renting them out, but it is not directly related to energy efficiency.

3.2.2 Energy Use Intensity (EUI)

As essential benchmarks of today's buildings, the site energy use intensity (EUI: kWh/m²), annual carbon emissions (kgCO²/m²), and costs from operational energy use ($/m²) were estimated using the *ClimateStudio* plugin (2022), which has the industry-standard building energy modeling software, *EnergyPlus* (2022) developed by the US Department of Energy as its backend engine. We use the same settings consistently across all layouts using conventional materials for residential surfaces, and the analysis area for the glazing on each surface was set to 80 percent for all exterior surfaces. The total annual site EUI for the entire building was calculated based on heating, cooling, lighting, and equipment.

3.2.3 Construction Cost Estimation

We estimated hard construction costs using US National Average costs from RSMeans cost data (2014) based on the method introduced in RSMeans (2014) and Mubarak et al. (2020). The national medians for hard construction cost per square footage are classified based on building floor height, i.e., low rise (1-3), mid rise (4-7), and high rise (8 and above), and were modified with a project size modifier from 2015 RSMeans data (2014). We calculated the costs for our layouts.

3.2.4 Visibility Analysis based on 2D Isovist (View)

The visibility of exterior spaces has a significant impact on the architectural and financial quality of each housing unit. As a result, each unit must be evaluated separately (Figure 3). We used the *DeCordingSpaces* plugin (2021) for *Grasshopper* to perform a single 2D *Isovist* calculation (Benedikt 1979, Batty 2001) from the centroid of each unit's area. As our interest is in the comparative study, we evaluated the maximum possible visibility by assuming that all exterior wall areas are transparent consistently across all layouts. We set the range (visible distance) to 800 feet and the view angle to 270 degrees, pointing away from the adjacent hallway and using interior and exterior walls as obstacles, as well as surrounding buildings. A 2D *Isovist* of each unit is simply quantified as the viewable area in square feet from each unit, which varies across units based on exterior conditions, as shown in Figure 3 (left). Higher the viewable area, the better is the value of the unit in our system. The mean of the 2D *Isovist* calculation results from all units was then calculated as a general value that indicates the overall quality of views for a given building layout.

3.2.5 Solar Radiation Analysis

The *Ladybug* plugin, which uses the *Radiance* (2022) and *EnergyPlus* backend engines, was used to analyze solar radiation across the entire exterior surface of each building. We looked at the total radiations in kWh for Cooling and Heating Degree Days, using 23.3°C and 18.3°C as the base temperatures, respectively. To assess the solar exposure, we set the value of total radiations during the Heating period subtracted by the Cooling period as an approximate guideline value. In the temperate climates under consideration in Figure 5, we assumed that the "higher the better" value is the case.

3.2.6 Courtyard Analysis (Garden)

The courtyard is a valuable shared space for residents, providing reasonable separation from public urban spaces and cultivating a sense of belonging and community among tenants. An annual solar radiation simulation on courtyard space was performed using the Ladybug plugin, which uses the *Radiance* and *EnergyPlus* backend engines to measure the building's overall courtyard solar performance, to assess the quality of a courtyard. We discovered that small footprints, relatively high unit requirements, and building orientation all have a significant impact on the amount of sunlight that enters the courtyard. In addition, we compared their square footage as potential green, open space areas (Figure 3).

In addition, we calculated the volume to surface area ratio known as Compactness, which is typically "larger is better" for lower energy consumption (BSC 2012). The sum of unit deviation amounts from the user-defined target numbers for the total unit count and three different unit types were also calculated based on the target percentages (Units).

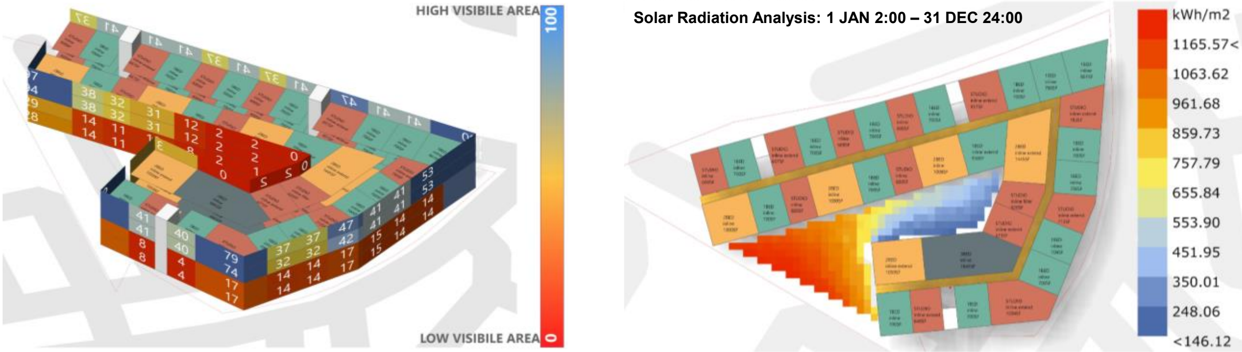

Figure 3: View Analysis based on 2D Isovist for each unit with a normalized score ranging from 0 to 100 (left), solar radiation analysis of a courtyard during heating degree days (right).

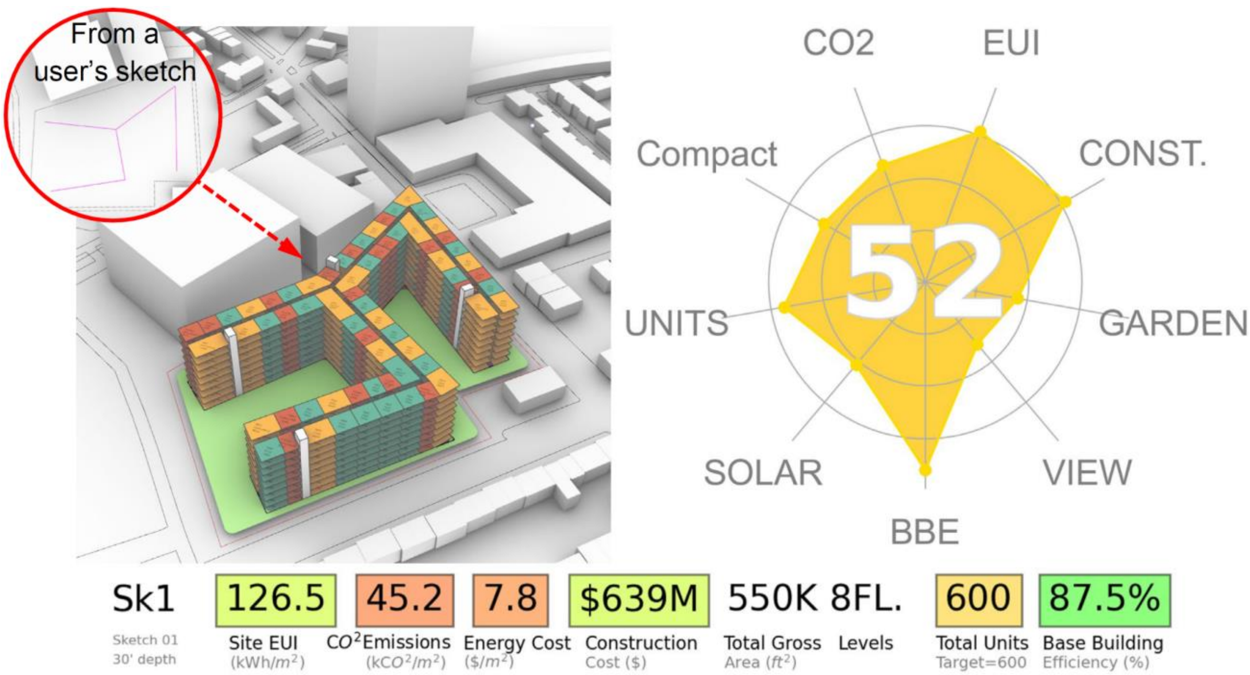

Figure 4: A generated layout for a housing complex based on a user's sketch (left) with a radar chart for nine criteria with an overall score (right) as described in Section 3.2 and values from the analysis used for the radar chart before normalization (bottom).

3.2.7 Rader Charts for the Performance Criteria

Before beginning to sketch, our program currently generates separate files for users to view results from up to 52 auto-generated layouts, including perspective images of buildings and performance evaluations on selected criteria with radar charts. For our radar charts, we selected nine criteria: BBE, Site EUI, CO^2 Emissions, estimated hard Construction Cost, Compactness, View Areas, Unit Count Deviation, Solar Radiation, and Courtyard Quality (Garden). Each criterion's score is normalized in the range of 0 to 1 for all 52 layouts, and these values are set to be "higher is better." We believe that higher values for Site EUI, CO^2 Emissions, Construction Cost, and Units' Deviation are undesirable in general, so we reversed the signs of these values before normalization. For the courtyard, we believe that larger potential areas for green garden and sufficiently higher solar exposure during heating degree days would be preferable in temperate climates, so the sum of normalized values from the area and solar radiation from the heating period were normalized once more to obtain a single value representing the garden's quality. Following that, we averaged the scores from the nine criteria for each layout and used this value as a guideline value to estimate the overall score of each layout (Figure 4 and 5).

Figure 5: Examples of generated performance evaluation datasets using our program's auto-generated pre-defined built forms. Each example shows a perspective of a building layout (left), a radar chart for nine criteria with an overall score (right), and values from the analysis used for the radar chart prior to normalization (bottom). (Notes: The results are based on the test site in the temperate climate zone on the East Coast of the United States, with the conditions set based on 600 units with percentages for unit types of 20% studio, 36% 1-bedroom, and 44% 2-bedroom apartments).

4 RESULTS

Our selected geometries are simple yet essential enough for users to grasp the relationships between forms and performance characteristics intuitively, and they can begin to incorporate their understanding into their sketches. Figure 6 shows the distribution of 52 layouts based on their performance scores in nine criteria from radar charts as their features using Principal Component Analysis (PCA) (Jolliffe 2011). PCA is a dimensionality reduction technique that arranges layouts in a 2-D plot with similar features (performance characteristics) closer together. We found that a similar distribution can be obtained with the t-Distributed Stochastic Neighbor Embedding (t-SNE) (Van der Maaten and Hinton 2008) with the perplexity set to 30. Since our interest is in observing a distance representing similarity among layouts, we only show the result of PCA, which has a higher cumulative contribution rate of the original data. Similar built forms are plotted closer together, implying that similar forms perform similarly, as can be seen from the ID tags of markers such as "Twr" (i.e., Tower) in Figure 6. It is important to note that our pre-defined layouts are generated parametrically by going through the options evenly separated from each other, for example, gradually from tall, medium, to low in height, and some characteristics cannot be discretely labeled or separated.

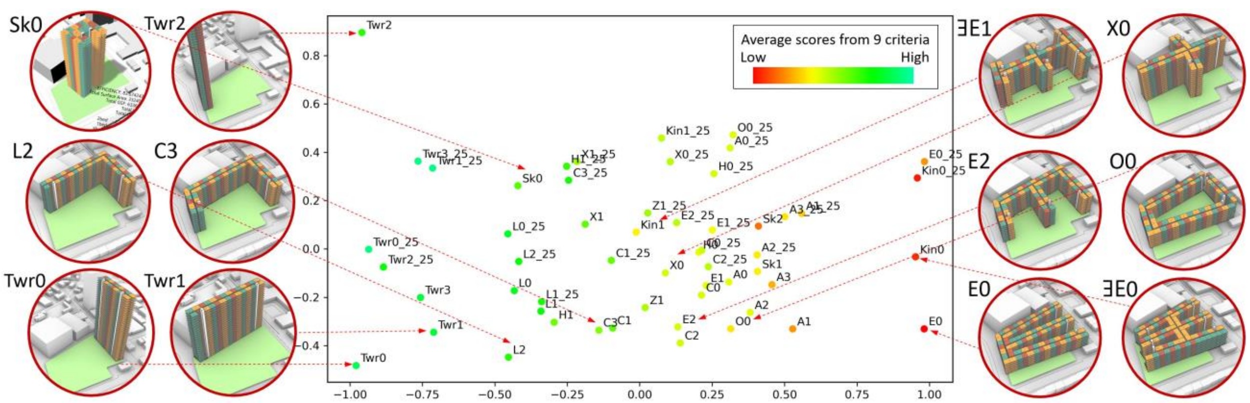

Figure 6: PCA visualizations of 52 layouts based on their performance scores in nine criteria from radar charts as their features. The color of a marker is based on the average score of all criteria from each layout (green being higher and red being lower).

There are strengths and weaknesses in each layout. As shown in the radar charts in Figure 5, towers distributed closer to the left side of the plot in Figure 6 have higher overall scores, including Site EUI, CO^2 emissions, views, and garden areas. However, because they have more floors, their BBE and construction cost scores are lower. The overall scores for "L," "C," and "O" shaped layouts are moderately good, with higher BBE and construction scores but lower view scores than towers. Except for relatively higher BBE and construction cost scores, extremely clumped layouts such as "E0" and "Kin0" (∃E shape) plotted closer together at lower right show significantly poor overall performance. These are denser solutions with potentially higher percentages of rentable spaces on a site with height-related zoning constraints. However, in terms of EUI, they are not eco-friendly solutions and provide a poor experience for residents. These more compressed layouts also have higher surface-to-volume ratios and lower compactness scores, resulting in lower solar radiation and visibility scores.

We learned from this project that there is a trade-off between the degree of freedom for outputs based on user inputs and the level of automation and optimization of the system. While some advanced multi-objective optimization platforms can provide users with more optimized and selected solutions based on building requirements and conditions, our platform allows for more potential controls for variations in outcomes, including some footprints based on users' sketches. According to our preliminary findings based on the experiences of four students who used our platform with the same input settings as shown in Figure 5, they all claimed to have used the tool iteratively and quickly to redraw their floor plans in order to obtain layouts that satisfied them in both performance and form. While our system does not optimize the footprint of the building from the start, it stimulated participants' motivation and creativity, allowing them to search for schemes that satisfy their aesthetics and performances in a balanced manner based on their post-experiment reports. To thoroughly evaluate the platform, however, a more extensive usability study is required.

5 LIMITATIONS

While our project allows users at all levels to sketch and view multiple built outcomes and results, the floor plans generated by our program are still in the preliminary stage and do not include spaces for elevator cores and mechanical rooms. Adding features that enable users to create custom conditions for each floor would allow users to consider more complex geometries for their designs, such as void openings and cantilevered volumes. Despite the fact that our EUI results are precisely estimated using reliable backend engines, the latency caused by EUI analysis for multiple examples prior to the user's sketching is currently not ignorable (i.e., up to 6 minutes each for options in Figure 5). The overall score for a layout is the mean of the normalized scores of nine criteria, which can have more personalized weights for each criterion rather than treating all weights equally. Users must also open separate files to view results from pre-defined basic layouts on our platform. We need to prioritize selecting specific layouts and appropriate measures for the preliminary design study.

6 CONCLUSION

We proposed an intuitive platform for users to visually study potential housing complex designs and their potential performance in EUI, environmental, and some financial criteria based on simple sketches created by users. Before users begin sketching, our program can generate up to fifty basic layouts with performance results. As a result, users can consider what they see for their new design by learning the intrinsic relationships between built forms and performance characteristics. Our analysis of 52 primary built forms also revealed the existence of relationships between built forms and performance characteristics in 9 criteria. They are valuable resources for preparing users to make their buildings more environmentally friendly while also allowing for some qualitative and exploratory geometrical changes by iteratively revising their designs using the sketching interface and quickly translating their preliminary geometries into computational evaluation models. While there are emerging high-performance façade materials and systems that could improve buildings' energy performance regardless of their relationships to external environments, the proposed system aims to help architects foster a better understanding of fundamental relationships between built forms and their performance at early exploratory stages, which could then be used to optimize the eventual outcomes of architectural development more efficiently at later stages.

It is debatable whether all auto-generated layouts should strictly follow the specific local codified regulations for the site under consideration by a user. As the goal of our project is to help professionals, including beginning architects, get a better sense of how a wide range of geometric variations commonly used around the world affects performances, we made sure to include massing options that are educationally and conceptually valuable. Our current interface necessitates that the user have a PC pre-installed with the necessary tools and plugins. It is ideal for running our tool directly on a browser using a remote server with the *Rhino.compute* API, allowing us to conduct a usability study with more participants and gather feedback for the assessment and improvement of our tool. Future work will also include more customization of our sketching interface based on feedback from more extensive future usability studies. Our current interface employs a drawing feature based on polygonal segments, allowing the CAD software to accept mouse input. Including a spline curves feature, allowing users to design more organic geometries, and incorporating different input devices, such as a digital stylus directly on a tablet screen, could all be considered.

ACKNOWLEDGMENTS

This research has been partially supported by the Hillier College of Architecture and Design Seed Grant award from New Jersey Institute of Technology (NJIT) and the National Science Foundation I-Corps site at NJIT award.

REFERENCES

Autodesk Inc. 2021. "Dynamo". https://dynamobim.org. Accessed Jan. 20, 2022.

Batty, M. 2001. "Exploring isovist fields: space and shape in architectural and urban morphology". *Environment and planning B: Planning and Design* vol. 28 (1), pp. 123–150.

Benedikt, M. L. 1979. "To take hold of space: isovists and isovist fields". *Environment and Planning B: Planning and design* vol. 6 (1), pp. 47–65.

Building Science Co. 2012. "BSD-061: The Function of Form—Building Shape and Energy". https://www.buildingscience.com. Accessed Jan. 20, 2022.

Chan, E. H., B.-s. Tang, and W.-S. Wong. 2002. "Density control and the quality of living space: a case study of private housing development in Hong Kong". *Habitat international* vol. 26 (2), pp. 159–175.

Christodoulou, A., M. Vola, and G. Rikken. 2018. "Case study for the application of multidisciplinary computational design assessment and constructability optimisation tools". In *Proceedings of the Symposium on Simulation for Architecture and Urban Design*, pp. 1–8.

DecodingSpaces 2022. "DecodingSpaces: Toolbox for Grasshopper". https://toolbox.decodingspaces.net. Accessed Jan. 20, 2022.

Depecker, P., C. Menezo, J. Virgone, and S. Lepers. 2001. "Design of buildings shape and energetic consumption". *Building and Environment* vol. 36 (5), pp. 627–635.

Do, E. Y.-L., and M. D. Gross. 2001. "Thinking with diagrams in architectural design". *Artificial Intelligence Review* vol. 15 (1), pp. 135–149.

EnergyPlus 2022. "EnergyPlus™". Accessed Jan. 20, 2022.

Fisher-Gewirtzman, D., and N. Polak. 2019. "A learning automated 3D architecture synthesis model: demonstrating a computer governed design of minimal apartment units based on human perceptual and physical needs". *Architectural Science Review* vol. 62 (4), pp. 301–312.

Gerber, D. J., S.-H. E. Lin, B. P. Pan, and A. S. Solmaz. 2012. "Design Optioneering: Multi-Disciplinary Design Optimization through Parameterization, Domain Integration and Automation of a Genetic Algorithm". SimAUD '12. San Diego, CA, USA, Society for Computer Simulation International.

Hastings, W. K. 1970. *"Monte Carlo sampling methods using Markov chains and their applications"*.

Jolliffe, I., 2011. *"Principal component analysis"*. Springer.

Kahn, L. I. 1931. "The value and aim in sketching". *T-Square Club Journal* vol. 6 (19), pp.18–21.

Kirkpatrick, S. 1984. "Optimization by simulated annealing: Quantitative studies". *Journal of statistical physics* vol. 34 (5), pp. 975–986.

Ladybug Tools LLC. 2021. "Ladybug Tools". https://www.ladybug.tools. Accessed Jan. 20, 2022. Levitt, D. 2012. *The housing design handbook: A guide to good practice*. Routledge.

Martello, S., and P. Toth. 1990. "Bin-packing problem". *Knapsack problems: Algorithms and computer implementations*, pp. 221–245.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. 1953. "Equation of state calculations by fast computing machines". *The journal of chemical physics* vol. 21 (6), pp. 1087–1092.

Mubarak, S. A. et al. 2020. *How to estimate with RSMeans data: basic skills for building construction*. John Wiley & Sons.

Narahara, T., and Yamasaki, T. 2019. "A Preliminary Study on Attractiveness Analysis of Real Estate Floor Plans". In *2019 IEEE 8th Global Conference on Consumer Electronics (GCCE)*, pp. 445–446. IEEE.

Narahara, T. and Terzidis, K. 2006. "Multiple-constraint Genetic Algorithm in Housing Design". In *Proceedings of the Association for Computer-Aided Design in Architecture (ACADIA) International Conference, Synthetic Landscapes, Digital Exchange,* pp. 418-425.

Narahara T. 2010. *"Self-Organizing Computation: A Framework for Generative Approaches to Architectural Design"* (Ph.D. thesis), Harvard University, USA, pp. 1–318, https://dl.acm.org/doi/book/10.5555/2395396.

Olsen, L., F. F. Samavati, M. C. Sousa, and J. A. Jorge. 2009. "Sketch-based modeling: A survey". *Computers Graphics* vol. 33 (1), pp. 85–103.

Pena, W. M., and S. A. Parshall. 2012. *Problem seeking: An architectural programming primer*. John Wiley & Sons.

Radiance-online.org 2022. "Radiance: A Validated Lighting Simulation Tool". https://www.radiance-online.org. Accessed Jan. 20, 2022.

RSMeans 2014. *RSMeans Open Shop Building Construction Cost Data 2015*. R S Means.

Schwartz, M. 2021. "Human centric accessibility graph for environment analysis". *Automation in Construction* vol. 127, pp. 103557.

Sidewalk Labs. 2021. "Delve empowers real estate teams to design better, faster, with less risk". https://www.sidewalklabs.com/products/delve. Accessed Nov. 20, 2021.

Solemma LLC. 2022. "ClimateStudio: Advanced daylighting, electric lighting, and conceptual thermal analysis.". https://www.solemma.com/climatestudio. Accessed Jan. 20, 2022.

Spacemaker 2022. "Spacemaker, an Autodesk product". https://www.spacemakerai.com. Accessed Jan. 20.

Steadman, P. 2014. *Building types and built forms*. Troubador Publishing Ltd.

TestFit, Inc. 2022. "Solve deals instantly with TestFit". https://testfit.io. Accessed Jan. 20, 2022.

Titulaer, R., A. Christodoulou, and M. Vola. 2019. "Multi-objective parametric massing optimisation for a sustainable and healthy urban living environment". In *Design Modelling Symposium Berlin*, pp. 737–751. Springer.

Van der Maaten, L., and G. Hinton. 2008. "Visualizing data using t-SNE.". *Journal of machine learning research* vol. 9 (11).

Weytjens, L., and G. Verbeeck. 2010. "Towards 'architect-Friendly' Energy Evaluation Tools". In *Proceedings of the 2010 Spring Simulation Multiconference*, SpringSim '10. San Diego, CA, USA, Society for Computer Simulation International.

AUTHOR BIOGRAPHIES

ZHONGMING PETER ZHANG is a Student in the Hillier College of Architecture and Design at New Jersey Institute of Technology. He recently received the ARCC King Student Medal for Excellence in Architectural and Environmental Design Research. His research interests include computation technology applications in the AEC industry. His email address is zpz2@njit.edu.

TARO NARAHARA holds a Doctor of Design from Harvard University and a Master of Science from Massachusetts Institute of Technology. He is currently an Associate Professor in the Hillier College of Architecture and Design at New Jersey Institute of Technology. He was awarded the 2020 Human Communication Award by IEICE Japan. His research interests include AI applications in architecture. His email address is narahara@njit.edu.