SYMPOSIUM ON SIMULATION IN ARCHITECTURE + URBAN DESIGN
SimAUD 2020

SpaceAnalysis: A Tool for Pathfinding, Visibility, and Acoustics Analyses in Generative Design Workflows

Rhys Goldstein[1], **Simon Breslav**[1], **Kean Walmsley**[1] **and Azam Khan**[2]

[1]Autodesk Research, Toronto, Canada, {firstname.lastname}@autodesk.com
[2]University of Toronto, Toronto, Canada, azam.khan.ca@acm.org

ABSTRACT

A growing number of architectural design efforts are making use of spatial metrics that characterize the experience of people in built environments. Metrics can make qualitative experience-related factors quantitative, and thereby assist in the exploration of a parametric or generative design space. To facilitate the adoption and development of generative design workflows, we introduce a tool called SpaceAnalysis that performs pathfinding, visibility, and acoustics analyses from which a variety of metrics can be computed. A theoretical contribution arising from this work is a new discretization method that converts 2D building geometry into a grid-based data structure supporting all three types of analyses. Experimental results show that the new method accommodates narrow corridors and small doorways with an efficient grid resolution of about 25 cm. We apply SpaceAnalysis to recreate and make publicly available a generative design workflow that was previously used to lay out a 250-person office.

Author Keywords

Grid-based methods, discretization; Dijkstra's algorithm; isovists; sound simulation, transmission-line matrix method.

ACM Classification Keywords

J.6 COMPUTER-AIDED ENGINEERING (Computer-aided design); G.2.2 DISCRETE MATHEMATICS: Graph Theory

1 INTRODUCTION

Human comfort, productivity, and wellness are greatly influenced by the design of the buildings and public spaces in which we spend the vast majority of our lives [11]. Recent investigations into the visual [16, 17] and acoustic [1, 15] implications of indoor space geometry are among many research efforts aimed at improving peoples' day-to-day experiences in built environments. Several spatial metrics characterizing a human experience—including adjacency, distraction, views to outside, and daylight, as shown in Figure 1—were incorporated by Nagy et al. [13] into a generative design workflow used to lay out a two-story office in Toronto for 250 knowledge workers. The idea behind the approach was to explore a diverse set of design options that would likely perform well from an office worker's perspective.

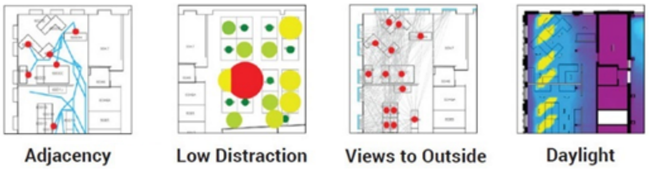


Figure 1: Four of the metrics characterizing human experience in the generative design work of Nagy et al. [13].

Our goal is to facilitate those who wish to adopt and further develop generative design workflows similar to the one pioneered for the Toronto office. To this end, we developed a tool called SpaceAnalysis that performs pathfinding, visibility, and acoustics analyses from which other experience-related metrics can be computed. Pathfinding is vital for adjacency calculations. It can also be used to identify congestion areas, a source of inconvenience and distraction yet also a potentially desirable element of a productive social environment [14]. Visibility is useful for quantifying both unwanted visual distraction and desirable views to the outside. Acoustics results enable privacy and noise-related distraction metrics.

The development of the SpaceAnalysis tool led to a number of theoretical contributions including a new discretization method and novel pathfinding and visibility algorithms inspired by Pascal's triangle. The discretization method converts line segments representing walls or other building elements into a space lattice data structure on which all three types of analyses are based. The use of a single grid-based data structure simplifies both the implementation and use of the tool, as a single geometry conversion function is all that is needed for pathfinding, visibility, and acoustics. Some precision is lost in converting building geometry into a grid, but our priority is computational speed. It is understood that for generative design workflows, the results need only be "good enough" to compare different options. Separate, precision-oriented tools can be used to fine-tune the final design.

SpaceAnalysis takes the form of a package for Dynamo [3], a visual programming environment. After presenting the discretization method and describing the three types of analyses, we present an open source recreation of the generative design project of Nagy et al. [13] that makes use of the new tool.

SimAUD 2020 May 25-27 Vienna, Austria

2 RELATED WORK

The features of SpaceAnalysis were guided by the Toronto office generative design project [13], which was itself an application of research in space layout generation, pathfinding, space syntax, and multi-objective optimization.

Space layout generation is a part of many generative design efforts in the architectural domain. Numerous layouts were generated for the Toronto office using Voronoi patterns controlled by a set of seed points. A variety of other geometry generation approaches are reviewed by Du et al. [6]. SpaceAnalysis focuses on the process that comes after space layout generation: the analysis of the generated design options.

Pathfinding is the computational process of finding shortest paths in environments such as buildings or urban neighborhoods. It enables a variety of metrics related to travel: the time and energy to walk from point A to point B, the distraction caused by people moving, unwanted crowding, but also the health and social benefits of passing through pleasant and engaging spaces. The simplest pathfinding methods are Dijkstra's algorithm [5] and the A* ("A-star") algorithm [7] applied on a regular grid of points. The method used in the Toronto project belongs to this category. Pathfinding is a heavily researched field, however, and as described in a review by Algfoor et al. [2] there are methods that use navigation meshes instead of grids, there are "any-angle" grid-based methods where paths are not constrained to grids, and there are hierarchical methods designed to efficiently handle large environments. Despite extensive research in the field, relatively little attention has been given to the discretization method by which travel barriers are converted to a grid representation. As will be shown in Section 3, the discretization method affects the spatial resolution of the grid, which in turn affects the speed of the pathfinding process.

Space Syntax [8] refers to a research community and its large collection of spatial analyses that have been pioneered and applied to architectural and urban design projects. Their contributions include isovists [18]: spatial geometries representing the area that is visible from a single point. Isovists were used in the Toronto project to evaluate workers' views to the outside and the extent to which they might be distracted by other occupants. While most visibility analyses of this nature rely on vector-based calculations, the visibility computation in SpaceAnalysis is unusual in that it employs a discrete, grid-based approach with no rays and no intersection tests.

Multi-objective optimization, the exploration of design options and their trade-offs, is the next step in a generative design workflow. An example of the technique is provided by Keough and Benjamin [12], who optimize both structural and material objectives in a context where the aesthetics of the design is of paramount concern. Fundamental to this approach is the idea that no single design performs best on every measure. It is therefore neither practical nor desirable to have the computer choose the final design. The designer still makes the decisions, but does so with access to numerous relatively well-performing generated options. Metrics computed using SpaceAnalysis can be optimized in this fashion using a separate Dynamo extension called Project Refinery [4].

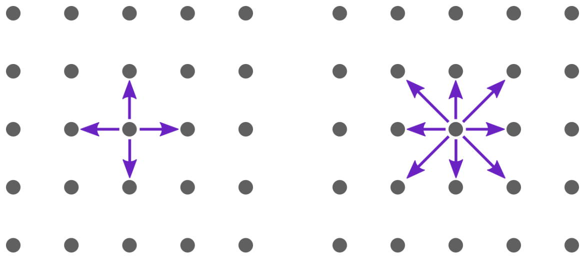
Acoustics analyses were not included as part of the Toronto office project, but promise to enhance future generative design efforts by allowing noise and privacy considerations to be part of the process. One way to simulate acoustics is the Transmission-Line Matrix (TLM) method [9], a simple algorithm that propagates impulses on a grid. If cell A transmits an impulse of 1 to cell B, then in the next time step cell B radiates impulses of $\frac{1}{2}$ in three directions and reflects an impulse of $-\frac{1}{2}$ back to cell A. Huang et al. [10] provide an example of indoor sound propagation using the TLM method.

3 SPACE LATTICES

The design, implementation, and use of the SpaceAnalysis package was greatly simplified by the decision to employ a single data structure for all types of supported analyses. The downside of this approach is that no single data structure is optimal for every analysis, and as a consequence we had to adapt some of our algorithms. Overall, the benefits of using a single data structure, which we refer to as a *space lattice*, outweighed the costs. Here we describe the structure of the space lattice, the discretization method used to construct it, and an experiment evaluating the discretization method.

3.1 Structure

The space lattice data structure is a 2D rectangular 8-neighbor grid of points. An 8-neighbor grid means that each point may be connected with its 8 closest neighbors on the axes and diagonals, in contrast to a 4-neighbor grid which excludes diagonal connections. Figure 2 illustrates both types of grids.


(a) 4-neighbor grid

(b) 8-neighbor grid

Figure 2: Illustration of (a) 4-neighbor and (b) 8-neighbor grids. The space lattice data structure of the SpaceAnalysis package is based on an 8-neighbor grid.

In the absence of barriers, all pairs of neighboring points are considered connected. In the case of pathfinding, this means a person at any point can travel in any of the 8 possible directions. Connections are bidirectional; if one can travel directly from one point A to a neighboring point B, then one can travel directly from B to A. Barriers such as walls are represented by severing connections between neighboring points.

Figure 3 shows a Dynamo node supplied by the SpaceAnalysis package to create a space lattice from an outer bounding box, a list of line segments representing barriers, and the grid resolution. The grid resolution is the spacing between neighboring grid points on the same axis—the shortest distance between points. Connections between points, shown in purple, are broken by a line at the bottom representing a barrier.

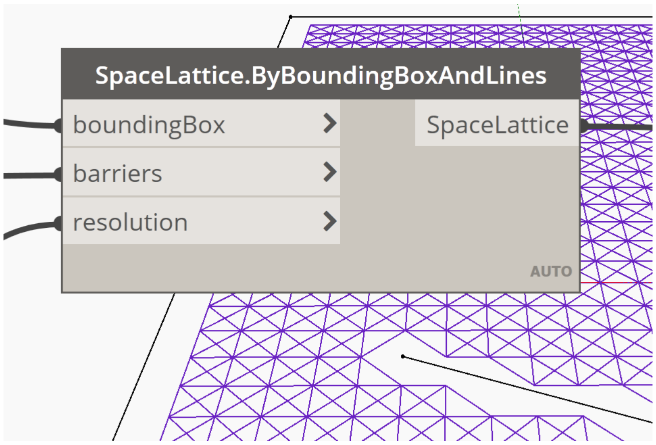


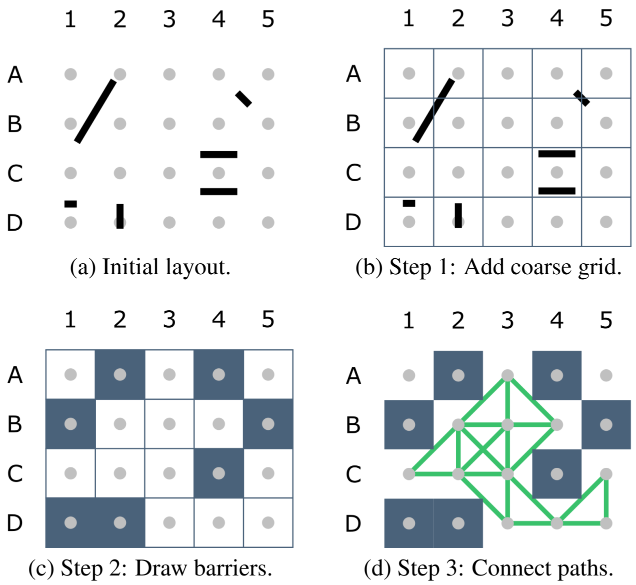
Figure 3: A Dynamo node that creates a SpaceLattice object from an outer bounding box, a list of line segments representing barriers, and the spacing between grid points.

3.2 Discretization

The line segments that represent barriers are primitive Dynamo objects and are not constrained to points on the space lattice grid. When these line segments are used to sever connections between grid points, they are effectively being converted from a continuous vector-based representation to a discrete grid-based representation. There are multiple ways to perform this discretization, and the method chosen may affect both the results of the various analyses and the appropriate choice of grid resolution. We implemented two discretization methods: a basic method and an enhanced method.

The ***basic discretization method*** in Figure 4 is the more obvious approach. The initial layout shows a 4x5 grid of points that may or may not be accessible depending on which connections get severed by six barriers represented by line segments (Figure 4a). The first step is to establish a cellular grid for processing the barriers (Figure 4b). In this case it is essentially the same as the original grid. The next step is to use Bresenham's standard line drawing algorithm to process the barriers by filling in grid cells as if they were pixels in an image (Figure 4c). The final step is to ensure the original points are connected to their neighbors wherever the path is not cut off by blocked cells (Figure 4d). Observe that the path from A1 to B2 is cut off by a pair of blocked cells that meet at a corner. The path from C1 to B2 touches the corner of a blocked cell, but is not cut off since one could skirt around.

The ***enhanced discretization method*** shown in Figure 5 differs from the basic method in two ways. First, the grid used to process the barriers is twice the resolution of the initial grid of points (Figure 5b). Second, after the barriers are processed using Bresenham's algorithm, blocked cells are expanded to fill their 4 nearest neighbors (Figure 5d), and then contracted by expanding the unblocked cells in the same manner (Figure 5e). These expansion and contraction steps fill in small gaps. The original points are then connected where possible. Notice that the path from B1 to C2 is cut off by a pair of blocked cells, but the path from C1 to B2 is traversable.



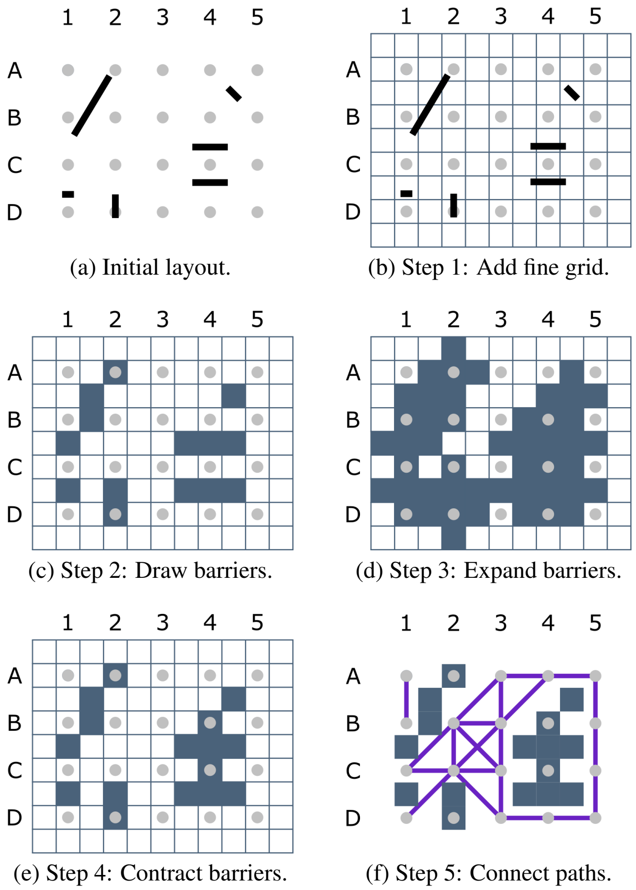
(a) Initial layout.

(b) Step 1: Add coarse grid.

(c) Step 2: Draw barriers.

(d) Step 3: Connect paths.

Figure 4: Basic discretization method with one resolution.



(a) Initial layout.

(b) Step 1: Add fine grid.

(c) Step 2: Draw barriers.

(d) Step 3: Expand barriers.

(e) Step 4: Contract barriers.

(f) Step 5: Connect paths.

Figure 5: Enhanced discretization method where barriers are draw at a finer resolution, then expanded, then contracted.

If we compare the outcome of the basic method in Figure 4d with that of the enhanced method in Figure 5f, we find that the same initial set of barriers can generate very different space lattices. For example, point B4 is accessible only when using the basic method, and point D1 is only accessible using the enhanced method. The enhanced method produces a route that encircles the barriers on the right-hand side of the environment, whereas the basic method eliminates this route.

The SpaceAnalysis tool uses only the enhanced method. This decision was based on the intuition that the enhanced method would more faithfully capture users' intentions when applied to real-world buildings and urban environments. It is worth noting that as long as any intended gaps between barriers are large compared with the resolution of the space lattice grid, then the choice of discretization method may not matter. The initial layouts in Figures 4 and 5 are somewhat contrived in that the gaps between the barriers are inadvisably small relative to the grid spacing. Nevertheless, it is helpful to investigate how the two methods handle small openings and narrow corridors where the grid resolution becomes important.

3.3 Experiment

Here we systematically compare the basic and enhanced discretization methods using the three scenarios shown in Figure 6. In each scenario, we are testing whether one can travel from point A to point B, which involves traversing a certain type of narrow passage. In the Wall Opening Scenario, the passage is a gap or doorway in the middle of a straight wall. In the Wall Junction Scenario there is also a gap or doorway, but the passage is situated at the corner of a T-junction formed by two walls. In the Corridor Scenario, the passage is the entire region between two parallel walls, a region that contains the two points.

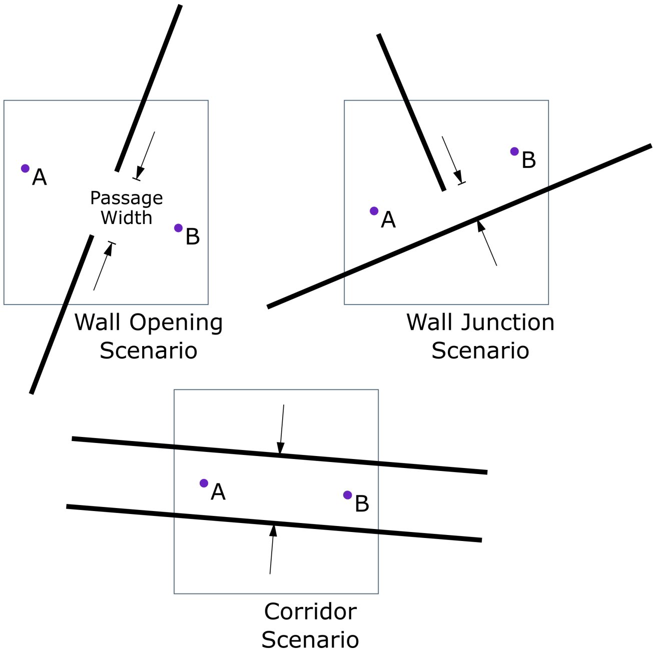


Figure 6: Scenarios used to compare discretization methods.

For each scenario, we varied the passage width from 0 to 4 at increments of 0.1, where the unit of measure is the resolution of the space lattice grid (i.e the grids shown in Figures 4a and 5a). To clarify, suppose the grid resolution is 25 cm. In that case, a passage width of 1 means the passage is 25 cm wide, and a passage width of 2 means the passage is 50 cm wide.

For each scenario and passage width, we ran 10000 tests to see whether one could travel from point A to point B. For each test, we rotated all of the geometry by a random angle, and translated it by a random displacement. Once the 10000 tests were run, we recorded the traversal rate: the fraction of tests where travel was successful. For extremely small passage widths, all routes from A to B were certain to be cut off regardless of how the scenario was rotated or translated. In those cases, all tests failed and the traversal rate was 0. For large passage widths, it was certain there would be a path from A to B through the passage. In those cases, all tests succeeded and the traversal rate was 1. But for certain passage widths in between, the orientation and placement of the geometry relative to the grid determined whether a route could be established or whether it was cut off. In those cases, the traversal rate was between 0 and 1.

The entire set of trials (3 scenarios × 41 passage widths × 10000 tests) was repeated for both the basic and enhanced methods. The results are plotted in Figure 7. For each scenario and discretization method, the curve has two notable passage widths: the minimum width where at least one test was successful; and the maximum width where all tests were successful. These width are listed in Table 1.

Scenario	Minimum Width		Maximum Width	
	Basic	Enhanced	Basic	Enhanced
Wall Opening	1.0	0.6	2.8	1.6
Wall Junction	0.6	0.8	3.0	2.3
Corridor	1.1	0.9	3.3	2.3

Table 1: Passage widths at which traversal is found to be possible (minimum width) and guaranteed (maximum width).

There are two criteria we should use to evaluate the methods. First, the incline of the curves in Figure 7 should be as steep as possible. Ideally the curves would be step functions, meaning that travel from A to B would depend only on the passage width and not on the rotation or translation of the geometry relative to the grid. The second criterion is that the incline should be as far to the left as possible. The further the incline is to the left, the coarser the resolution one can choose while still ensuring passage through small doorways and narrow corridors. Coarser resolutions mean fewer grid points, faster analyses, and potentially better experiences for users interactively exploring numerous designs options.

Based on these criteria, the enhanced method clearly outperforms the basic method in the Wall Junction and Corridor scenarios. The inclines are narrower and further to the left. With the enhanced method, a resolution of 25 cm essentially guarantees that travel paths will connect through passages of at least 60 cm. The basic method would require a finer resolution of 18 cm to ensure all 60 cm gaps are traversable.

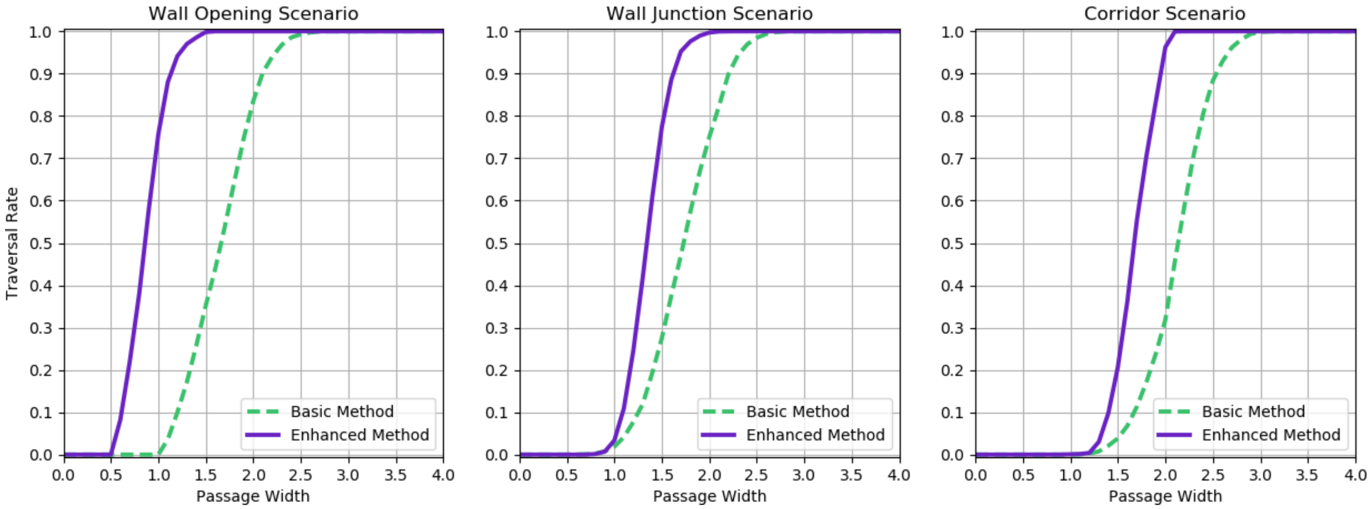


Figure 7: Traversal rates as a function of passage width for the three scenarios and two discretization methods.

The enhanced method does have a weakness, which is revealed by the Wall Opening Scenario. With a resolution of 25 cm, it is possible for a route to pass through a gap of only 15 cm. For applications involving human travel, this is too narrow. The steps of expanding and then contracting blocked cells in Figures 5d and 5e are intended to cut off these small gaps by causing a few accessible cells to become filled in. But this is only effective in situations like the Wall Junction and Corridor scenarios where the passage is bounded by the face of a wall. In the Wall Opening situation, those steps have no effect. Nevertheless, our perspective is that designers rarely create narrow vertical gaps with the intent of restricting travel. Furthermore, if the basic method were used with the necessary resolution of 18 cm or finer, then the Wall Junction tests show it is possible for a route to pass through a gap of only 11 cm. Thus we find that the experimental results validate our decision to use the enhanced method, and we can recommend a space lattice resolution of 25 cm for analyzing human travel in environments with doorways of at least 60 cm in width.

4 ANALYSES

Three areas of focus—(1) how people travel through a space and what interactions arise as a result, (2) what people can see from various locations, and (3) what people can hear—account for much of the impact that a layout option can have on people's experiences. These are the areas addressed by the by SpaceAnalysis tool's analyses: (1) pathfinding, (2) visibility, and (3) acoustics. The analysis methods are chosen with the aim of providing rapid feedback at a level of quality sufficient for comparing a diverse set of design options. Here we describe the implementation and use of each analysis.

4.1 Pathfinding

Pathfinding is a principal feature of the SpaceAnalysis package. It can be used for visualizing travel routes and computing metrics that capture the adjacency of locations, desirable and undesirable congestion, and worker distraction. In addition, the code that supports pathfinding serves also as a foundation for the visibility calculations described in Section 4.2.

Paths in SpaceAnalysis are computed by applying Dijkstra's algorithm to the 8-neighbor space lattice grid. With this approach, one starts with a single point, then computes paths inward from the closest adjacent neighbors, then computes paths inward from the next closest set of neighbors, and so on until the processed region either fills the entire lattice or encompasses some other specified point. For simplicity, steps to nearest and diagonal neighbors have distances of 5 and 7.

To take advantage of Dijkstra's algorithm, the PathField node precomputes all paths to or from a single point. Figure 8 shows an example of this node in use. It is also possible to bypass the PathField node and compute a single route directly from a SpaceLattice object and a start and end point.

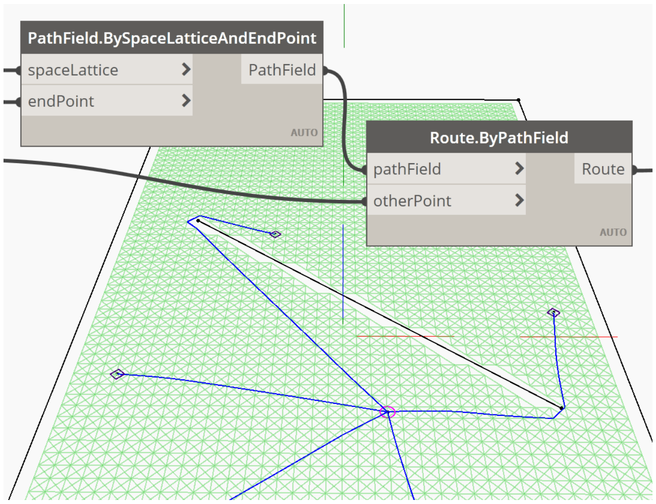


Figure 8: An example of pathfinding in SpaceAnalysis.

When finding paths on a grid, there is generally more than one shortest path between any two points. SpaceAnalysis aims to select the path that crosses empty regions with minimal deviation from the direct line-of-sight route. This is accomplished by assigning what we call a “Pascal number” to each point on any shortest path. Inspired by Pascal’s triangle, a Pascal number is the number of shortest paths that go through a point. Repeatedly selecting the highest Pascal number results in paths that tend to appear jagged, but can easily be smoothed into relatively straight curves. A “tidyIterations” parameter is provided to iteratively smooth jagged paths.

4.2 Visibility

Maximizing compelling views, minimizing visual distraction, and ensuring certain elements can be seen from certain locations are among the objectives that require visibility analysis. This feature of SpaceAnalysis takes advantage of the decision to use a single data structure for all analyses. Whereas most methods for computing visibility involve geometric projections or ray-object intersection tests, SpaceAnalysis uses a grid-based approach that re-purposes the implementation of Dijkstra’s algorithm developed for pathfinding.

Starting from a single view point, shortest paths are computed outward. By “outward”, we mean that the connection from any point A to a neighboring point B exists only if the trajectory of this step differs by at most 45 degrees from the vector originating at the view point and terminating at point A. In other words, all paths head as straight as possible outwards. Every visited point is assigned a Pascal number, the number of shortest paths to that point. To compute the point’s visibility, its Pascal number is divided by the Pascal number that would be obtained if there were no barriers. Hence the visibility from the view point to any other point is approximated as the fraction of shortest paths that are not cut off by barriers.

A Dynamo user performs the analysis by supplying parameters to a ViewField node, and then using a VisibilityGrid node to access the resulting array of fractions approximating visibility. An example with two barriers is shown in Figure 9.

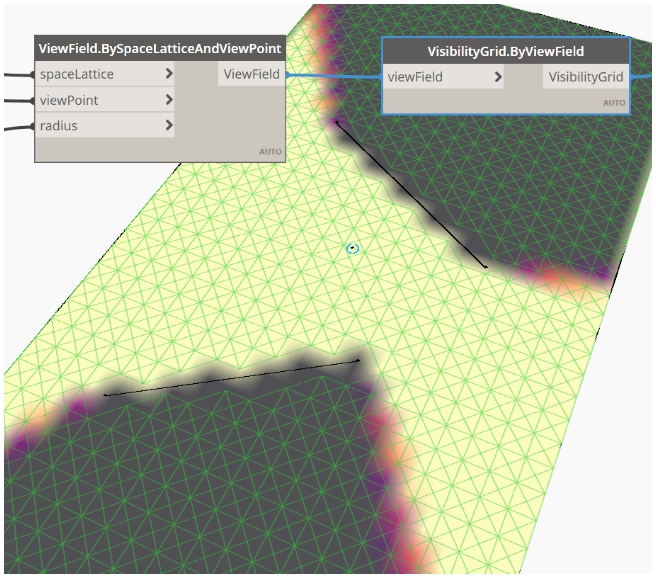


Figure 9: An example of visibility analysis. Surface colors indicate the region visible from a view point in the middle.

Two additional nodes assist with visibility analyses involving multiple points. One node creates a VisibilityGrid by taking the union of the results of multiple ViewField nodes. Technically the node reports the maximum visibility value of each grid point for all ViewFields. In essence, however, this can be interpreted as the union of a set of view fields: the region visible from any of the view points. The other node creates a VisibilityGrid from the intersection—or technically the minimum values—of multiple ViewField nodes: the region visible from all of the view points.

4.3 Acoustics

It can be desirable for a person at one location to hear sound from another, or it can be a privacy concern or source of distraction. The acoustics analysis supported by SpaceAnalysis provides a convenient way to evaluate these types of concerns. The tool simulates sound using the transmission line matrix (TLM) method described in Section 2. The basic TLM method assumes a 4-neighbor grid, which posed a challenge for us because the space lattice data structure is based on an 8-neighbor grid (see Figure 2). We therefore adapted the method to propagate sound impulses from a point to a diagonal neighbor, but only if barriers would prevent the impulse from reaching that neighbor via two axis-aligned steps (i.e. one step forward, one step sideways). The user interface for acoustic analysis is similar to that of the visibility analysis, with SoundField, AudibilityGrid, and associated union and intersection nodes. The node setup is illustrated in Figure 10.

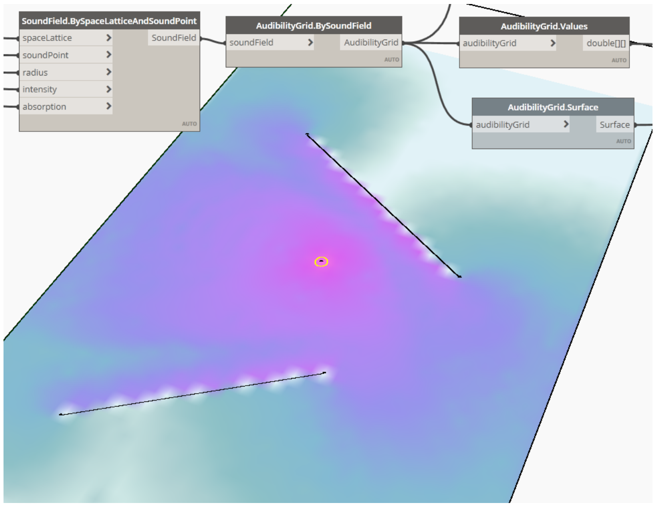
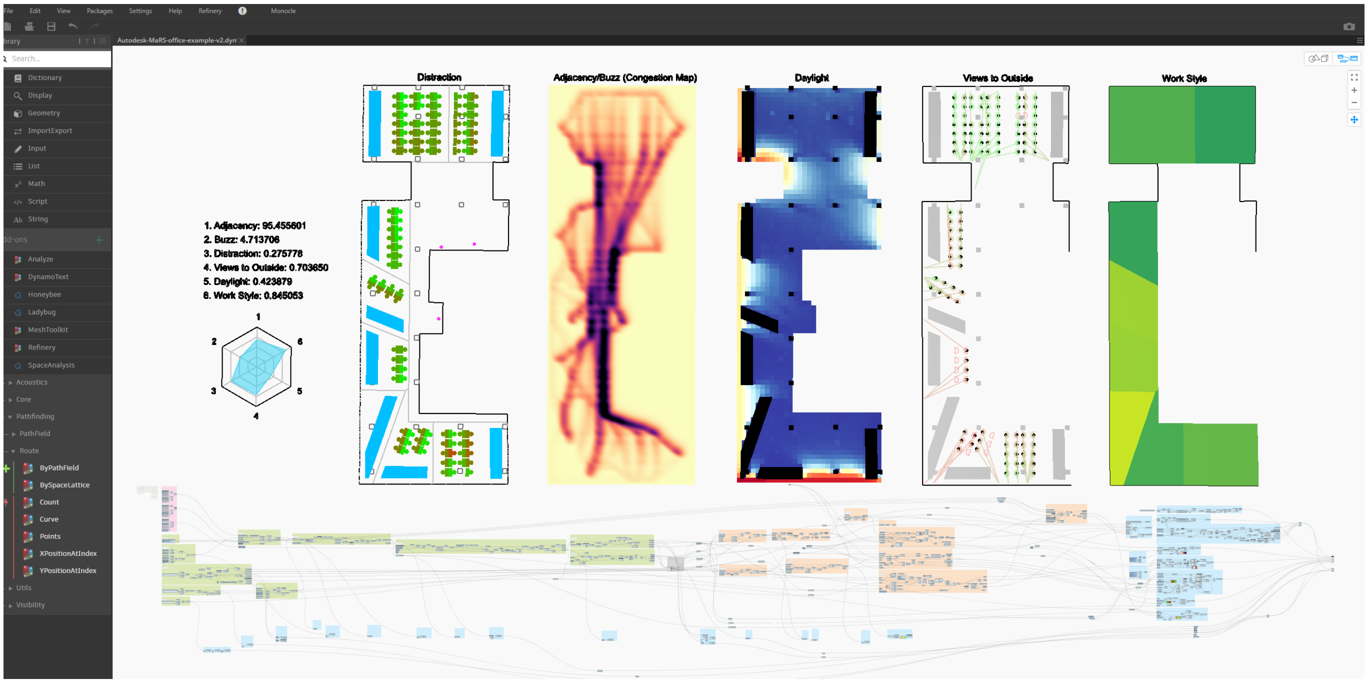


Figure 10: An example of acoustics analysis. Surface colors indicate the propagation of sound from a point source.

The sound simulation assumes single-frequency waves. This has the noticeable effect of causing interference patterns, particularly when the SoundSystem node is used to model multi-speaker arrangements. The wavelength of the sound can be obtained using the Wavelength node. In the current version of the tool, this wavelength is always $8\sqrt{2}$ times the space lattice grid resolution, which works out to 120 Hz for a resolution of 25 cm. The reason the wavelength-resolution ratio is fixed is to prevent degenerate results (small ratios) or long computation times (large ratios).



1. Adjacency: 95.455601
2. Buzz: 4.713706
3. Distraction: 0.275778
4. Views to Outside: 0.703650
5. Daylight: 0.423879
6. Work Style: 0.845053

Figure 11: The recreated Toronto office generative design project. Distraction, Buzz, Daylight, Views to Outside, and Work Style metrics are visualized at the top; the Dynamo graph is shown beneath. Most of the metrics were computed using SpaceAnalysis.

5 CASE STUDY

To demonstrate features of the SpaceAnalysis package, we have performed a recreation of Autodesk's generatively designed office in Toronto's MaRS Discovery District using Dynamo. The structure of the graph can be seen in Figure 11, where nodes can roughly be grouped into four categories: input parameters, geometry system, analysis, and visualization nodes. Input parameters are located on the left side of the graph inside pink groups and are defined by slider nodes. Geometry system nodes are inside green groups, analysis or metrics related nodes are inside orange groups, and visualization related nodes are inside blue groups.

The geometry system closely follows Nagy et al. [13], where we define a number of neighborhoods using Voronoi partitioning. Input parameters control the shape and size of neighborhoods as well as amenity locations within the neighborhoods. By manipulating input parameters, the geometry system is able to generate different layouts for the office, defining the solution space for the optimization search.

Each metric is aggregated into a single number to be used by the optimizer; however for visualization purposes, metrics are calculated at the individual desk or neighborhood level as seen in Figure 11. While Daylight and Work Style metrics did not require SpaceAnalysis, the package was used for Adjacency, Buzz, Views to the Outside, and Visual Distraction.

Adjacency characterizes the distance from desks to amenities such as meeting rooms, washrooms, or egress points. Once SpaceAnalysis finds the shortest path from each desk to each amenity, we average the distances to obtain an overall score. Lower scores indicate shorter, more desirable travel paths.

The ***Buzz*** metric used in our graph is slightly different from the original equation in Nagy et al. [14]. We define it as

$$Buzz = \frac{\sum_i d_i H_i}{\sum_i d_i h_i} \qquad (1)$$

where i represents a given path in a set of all adjacency paths, d_i is the length of path i, $H_i = \sum_j |c_{i,j+1} - c_{i,j}|$ with $c_{i,j}$ being the level of congestion at point j along path i, and $h_i = \max_j(c_{i,j}) - \min_j(c_{i,j})$. Congestion values $c_{i,j}$ are calculated by aggregating and smoothing all travel paths. Whereas the Buzz metric defined by Nagy et al. [14] characterizes the extent to which congestion areas are dispersed throughout a space, ours attempts to quantify the degree to which travel routes intersect one another. Designs achieving higher scores are seen as promoting interaction among people who frequent different areas of a productive social environment. We leave a more extensive validation and comparison of the metric for future work.

Views to Outside assesses visibility from each desk to the outside windows, giving seats closer to the windows a higher score and the ones further away a lower one.

Visual Distraction measures the amount of visual distraction people have at their desks. Distractions can come either from people at other desks or parts of the office that have high levels of congestion.

To cut down on the size of the graph, part of the implementation of the metrics is done using Python scripts embedded in Dynamo nodes. The Python code in these nodes can invoke functions corresponding with the nodes of the SpaceAnalysis package. Please refer to the source code of the case study for more details on how to use SpaceAnalysis from within Python scripts in Dynamo. The code can be downloaded from https://autode.sk/mars-graph. The SpaceAnalysis package itself can be installed from within Dynamo, by going to the Packages → Search for a Package menu item and searching for "SpaceAnalysis". As there have been over 2,000 downloads, the tool appears to be of interest and value to the computational design community.

In order to find the best designs according to the defined metrics, Project Refinery was used to perform multi-objective optimization. Besides optimization, Refinery supports exhaustive and randomized searching of a parameter space. After a set of design options are evaluated, the user can explore the results of the optimization using various plotting techniques such as scatter and parallel coordinate plots.

6 CONCLUSION

The types of spatial analyses, simulations, and metrics considered in this paper share a common goal of making qualitative experience-related factors quantitative, thereby assisting designers in the pursuit of more functional, productive, safe, healthy, visually compelling, and socially stimulating built environments. The presented tool contributes to that goal on both a theoretical and practical level. On the theoretical side, we have proposed a new discretization method for 2D building geometry, and shown that it accommodates narrow corridors and small doorways with an efficient grid resolution of about 25 cm. On the practical side, SpaceAnalysis supports those seeking to adopt and further pioneer generative design workflows for architecture and urban design.

The main strengths and weaknesses of the tool relate to the fact its underlying data structure is limited to two dimensions and a single scale. Future research will explore how a similar representation can support three dimensions and multiple scales, enabling movement on slopes and staircases, daylight analysis, acoustic simulations that account for room height, and projects with both building- and urban-scale elements.

REFERENCES

1. Alambeigi, P., Burry, J., and Cheng, E. Investigating the effects of the geometry on speech privacy of semi-enclosed meeting spaces. In *Proceedings of the Symposium on Simulation for Architecture and Urban Design*, SimAUD (2017).
2. Algfoor, Z. A., Sunar, M. S., and Kolivand, H. A comprehensive study on pathfinding techniques for robotics and video games. *Int. J. Comput. Games Technol. 2015* (2015).
3. Autodesk, Inc. Dynamo (`https://dynamobim.org`).
4. Autodesk, Inc. Project Refinery Beta (`https://www.autodesk.com/campaigns/refinery-beta`).
5. Dijkstra, E. W. A note on two problems in connexion with graphs. *Numerische Mathematik 1*, 1 (1959), 269–271.
6. Du, T., Turrin, M., Jansen, S. C., van den Dobbelsteen, A., and Biloria, N. A review on automatic generation of architectural space layouts with energy performance optimization. In *Proceedings of the International Conference On Building Energy & Environment*, COBEE (2018).
7. Hart, P. E., Nilsson, N. J., and Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics 4*, 2 (1968), 100–107.
8. Hillier, B., Leaman, A., Stansall, P., and Bedford, M. Space syntax. *Environment and Planning B: Planning and Design 3*, 2 (1976), 147–185.
9. Hoefer, W. J. R. The transmission-line matrix method—theory and applications. *IEEE Transactions on Microwave Theory and Techniques 33*, 10 (1985), 882–893.
10. Huang, P., Kapadia, M., and Badler, N. I. SPREAD: Sound propagation and perception for autonomous agents in dynamic environments. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA (2013), 135–144.
11. International WELL Building Institute (IWBI). WELL v2 pilot (Q3 2019), Accessed October 7, 2019 at `https://v2.wellcertified.com/v/en/overview`.
12. Keough, I., and Benjamin, D. Multi-objective optimization in architectural design. In *Proceedings of the Symposium on Simulation for Architecture and Urban Design*, SimAUD (2010).
13. Nagy, D., Lau, D., Locke, J., Stoddart, J., Villaggi, L., Wang, R., Zhao, D., and Benjamin, D. Project Discover: An application of generative design for architectural space planning. In *Proceedings of the Symposium on Simulation for Architecture and Urban Design*, SimAUD (2017).
14. Nagy, D., Villaggi, L., Stoddart, J., and Benjamin, D. The buzz metric: A graph-based method for quantifying productive congestion in generative space planning for architecture. *Technology—Architecture + Design 1*, 2 (2017), 186–195.
15. Peters, B. Integrating acoustic simulation in architectural design workflows: the FabPod meeting room prototype. *SIMULATION 91*, 9 (2015), 787–808.
16. Rockcastle, S., Ámundadóttir, M. L., and Andersen, M. A simulation-based workflow to assess human-centric daylight performance. In *Proceedings of the Symposium on Simulation for Architecture and Urban Design*, SimAUD (2017).
17. Turan, I., Reinhart, C., and Kocher, M. Evaluating spatially-distributed views in open plan work spaces. In *Proceedings of the IBPSA International Building Simulation Conference*, BS (2019).
18. Turner, A., Doxa, M., O'Sullivan, D., and Penn, A. From isovists to visibility graphs: A methodology for the analysis of architectural space. *Environment and Planning B: Planning and Design 28*, 1 (2001), 103–121.