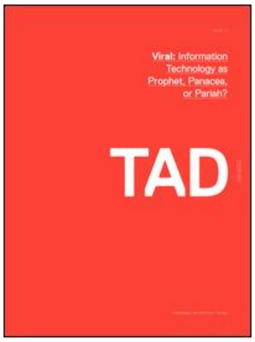


Technology|Architecture + Design

ISSN: 2475-1448 (Print) 2475-143X (Online) Journal homepage: http://www.tandfonline.com/loi/utad20

The Buzz Metric: A Graph-based Method for Quantifying Productive Congestion in Generative Space Planning for Architecture

Danil Nagy, Lorenzo Villagi, James Stoddart & David Benjamin

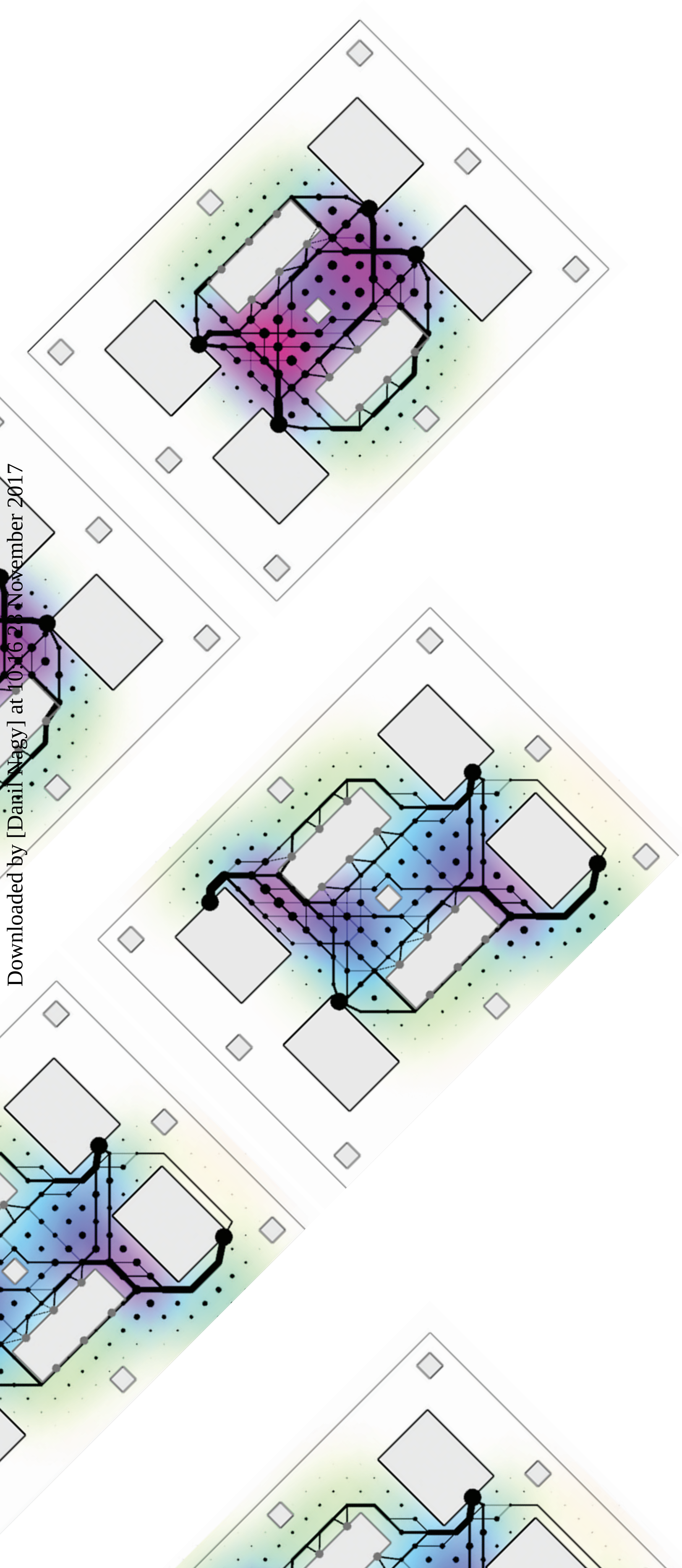
Danil Nagy
The Living, an Autodesk Studio

Lorenzo Villagi
The Living, an Autodesk Studio

James Stoddart
The Living, an Autodesk Studio

David Benjamin
The Living, an Autodesk Studio

The Buzz Metric: A Graph-based Method for Quantifying Productive Congestion in Generative Space Planning for Architecture



This paper describes a novel simulation method for measuring the amount of *buzz or productive congestion* in an interior architectural space. This "buzz metric" first predicts the amount and distribution of congestion based on a simulation of walking paths and, then, computes the extent to which this congestion is distributed across the space. Unlike dynamic methods for simulating occupant behavior such as agent-based crowd simulation, our method is computed statically, and thus is fast enough to be used in an automated generative design (multiobjective optimization) workflow. After describing how the metric is calculated, the paper demonstrates how it can be used in an automated generative design process through a case study of the design of an exhibit hall.

Introduction

Simulation in Generative Design

Generative design integrates technologies of artificial intelligence into the design process through the application of metaheuristic search algorithms to discover high-performing results within a given design system. This technology enhances a human designer's creative capacity by allowing them to explore many more designs than would be possible through manual methods. Since the algorithm has no intuition or prior bias for particular design approaches, these methods also have the potential to uncover novel design solutions that may not have been considered by the human designer.

The generative design framework is dependent on three main components. The first is a generative geometry model that defines a "design space" of possible design solutions. The second component is one or more measures that describe the objectives or goals of the design problem. Well-designed measures are a crucial part of the generative design process because they tell the computer how to evaluate each design iteration. Thus, the measures must capture as many of the important criteria as possible while still being directly calculable from the design space model. The third component is a metaheuristic optimization algorithm that can search through the design space to find a variety of high-performing design options by iteratively sampling possible solutions, evaluating them based on specified performance factors, and using this information to derive higher performing designs.

Simulation techniques are a critical part of the generative design process because they allow the optimization algorithm to digitally evaluate designs based on how they would perform in the real world. Two common simulation techniques used in mechanical engineering are finite element analysis (FEA), which simulates how forces flow through physical objects, and computational fluid dynamics (CFD), which simulates the motion of fluids such as liquid and gas. Although these physical simulations can also be applied to building design, such measures are more in the domain of engineering and are rarely the only or even the primary goals of an architectural design project. Even more important are occupant-level, "human" concerns such as how the space will be used, how the space feels, and how well the layout of the space matches the needs of the program.

To quantify such human-level metrics architects often rely on crowd simulation, which uses agent-based models to simulate the movement of people in a space. Such models can be extremely accurate, but since they must be calculated dynamically (over a series of time steps), they are also very computationally expensive, often taking several hours or even several days to compute a single design. Although this may be suitable to validate single design solutions, it is intractable in an automated generative design process that often involves the evaluation of thousands of design options before an optimal solution is found.

Productive Congestion

One occupant-level spatial quality that has generated interest in interior architecture and space planning is the concept of *buzz* or *productive congestion*. Congestion is often seen as a negative quality in architectural space as it can cause uncomfortable or even dangerous situations. However, in some public and work environments, a certain amount of congestion is useful for activating the space and creating productive interactions between occupants.

In office design this quality is often called "serendipity."[1] In a recent article regarding designing for serendipity in technology and creative offices, Silverman explains that "firms are thinking up new ways to encourage interactions among employees who normally don't work with each other [like] squeezing workers into smaller spaces so they are more likely to bump into each other."[2] The idea is that such curated moments of congestion can increase the "buzz" in an office space, making it not only a productive but an active, stimulating, and exciting place to work.

While it may not be obvious, such congestion can also improve productivity in an office, and the correlation between the two is well documented in the literature. Brown et al. suggest that "chance conversations [...] have long been judged to be essential for team coordination, cohesiveness and productivity."[3] Pentland et al.[4] and Haynes[5] provide further evidence to support the connection between congestion and office productivity, while Whittaker et al.[6] propose ways in which certain office layouts can boost informal collaboration. Waber, Magnolfi, and Lindsay also claim that "chance encounters and interactions between knowledge workers improve [individual and team] performance."[7]

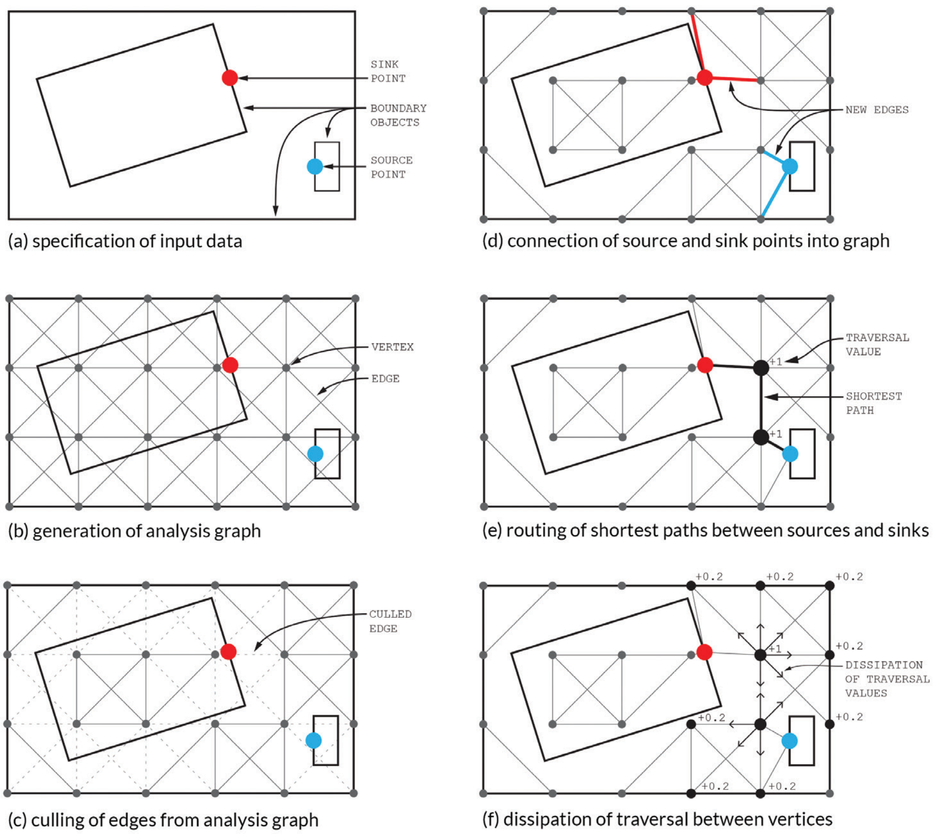
While the benefits of certain types of congestion have been extensively studied in office environments, encouraging active and lively spaces is also a crucial aspect in the design of public commercial spaces such as shopping malls and exhibit halls. In this case, pockets of congestion can create active zones that draw in more people to different areas of the space. This is crucial to the success or failure of such spaces, which rely on generating foot traffic for their shops and exhibitors.

Despite extensive research into the role of congestion in boosting productivity in office and public environments, there have been no concrete guidelines developed for ways spaces should be designed to encourage such productive congestion, and no specific measures which can be used to evaluate this quality within an architectural space. Architects designing such spaces usually rely on their experience, intuition, and rules of thumb to ensure that any congestion created in a space is productive and does not lead to uncomfortable or dangerous conditions for the building's occupants. For example, designers often centralize the locations of water coolers, coffee machines, and photo copiers to encourage spontaneous encounters between different groups of workers in an office. However, such heuristic approaches may miss design concepts and spatial layouts that support productive congestions but may be outside of the designer's intuition or prior experience. The goal of this paper is to support existing research into productive congestion, while offering a concrete methodology by which designers can evaluate and quantify this type of congestion within their architectural designs.

Related Work

The quantification of spatial experience is mostly concentrated in two fields: crowd simulation based on the dynamic interaction of agents, and space syntax, which is a collection of static methods for analyzing the physical characteristics of a space. Crowd simulation

▷ Figure 1. Computation process for buzz metric, ©2017 The Living, an Autodesk Studio.



is an extremely detailed and computationally expensive method, but yields highly accurate simulations for how people perceive and move around a space. Space syntax, on the other hand, uses static calculations to evaluate various topological and morphological features of a space. The benefits of this approach is that it can be evaluated very quickly. The results, however, tend to be very general, without being specific to program, human behavior, or yielding concrete metrics that can be used for automated generative design.

Crowd Simulation

Crowd simulation has received a great deal of attention in the recent decade, mostly having to do with its applications in computer graphics for simulating human figures in games and film. Crowd simulation has also been used in architecture for simulating evacuation scenarios[8] and for realistic scene rendering in urban planning and design.[9] However, a minimal amount of research in using behavior-based models for automated space planning and layout exists. This stems from the fact that such methods are typically computationally expensive and are, thus, not viable for generative design methods where many design options need to calculated.

Nevertheless, there have been several examples of crowd simulation techniques being used for the generation or evaluation of design spaces. Huerre et al.[10] simulated realistic crowd flow behavior based on computational models trained on real world data, which can run much quicker than traditional agent-based methods. Li and Liu[11] used agent-based and cellular automata (CA) models programmed with sustainable development strategies to automatically evaluate urban planning scenarios. Aschwanden et al.[12] combined procedural modeling of urban environments with an agent-based traffic system for automatically evaluating the generated models. The evaluation focused mostly on negative aspects of congestion, including traffic bottlenecks, stress levels, and exhaustion.

Another recent paper by Feng et al.[13] proposes to speed up the process of crowd simulation to the point of being useable for generative design by first abstracting the simulation process using a trained nonlinear machine learning model. This research is extremely promising because it can be extended to many other types of dynamic simulations beyond crowd simulation. The disadvantage of these methods, however, is that the resulting model is only an abstraction of the actual human behavior it is simulating, and in practice such models can be difficult to query or validate. Similarly to Feng et al.,[14] our method proposes a new way to statically compute occupant-level metrics such that they may be useable for generative design. However, our method maintains a direct relationship to the behavior of the occupants it is trying to simulate.

Space Syntax

Another method for quantifying the experience of architectural space is Space Syntax, a collection of static methods for computing quantitative metrics from the physical form of a space. The advantage of these methods is that they can be computed quickly and, thus, have been popular in generative design applications. The downside, however, is that most of their calculations are based only on analyzing the physical morphology and topology of architectural spaces, without taking occupant-level experiences or program-specific behaviors into account. Thus, they can only yield general metrics about the form of the space, without being specific to its program or occupation.[15]

One of the methods in Space Syntax that take occupant experiences into account is the axial map, which is often used to measure pedestrian activity in an urban context.[16] A related method is the visibility graph, which uses spatial graphs derived from isovists and viewsheds to consider perception issues such as accessibility and visibility in a space.[17] Penn and Turner[18] further this research by converting a visibility graph into a full agent-based model and then validating it based on real-world crowd observations. Finally, Stucky and Lee[19] describe a method for using viewsheds and isovists for determining optimal routes by finding shortest paths through implicit graphs.

Like many Space Syntax methods, our computation relies on a graph data structure, which is an extremely efficient way to compute useful statistics regarding a physical space. In our case, we use the graph not only to generate sample points from which to evaluate the space, but also to physically route simulated occupants through the space. Although our graph-based method is not as accurate as dynamic crowd simulations, it is also much faster since it can be computed statically in a single time step. This means that our method is fast enough to be integrated into an automated generative design workflow, while still being specific to the program and layout of the space.

Methodology

This section describes the process by which the buzz metric is calculated for a given 2-dimensional architectural floor plan (Figure 1).

Input Data Structure

The calculation requires three inputs (Figure 1a). The first input is a list of all the boundary objects in the floorplan as polygons. The second is a set of points that define the sources of the occupant movement, or where the occupants of the space are coming from (for example entrances to buildings or employee desks). The third is a set of points that define the sinks of occupant movement, or where the occupants are going to (for example doors to common areas in an office or shops in a mall).

Generating the Analysis Graph

The first step is the overlay of a traversal graph onto the floor plan. In general, a graph is a computational data structure G = (V, E) composed of a set of vertices (V) and a set of edges (E) that connect pairs of vertices together. Graphs have often been used for architectural, landscape, and urban analysis by assigning vertices to locations in space, and using edges to represent various types of relationships between them. In our case, the graph's vertices are evenly sampled within the bounds of the space, and the graph's edges describe walkable paths between two adjacent vertices. Given the boundary geometry, a graph is created such that:

1. The graph covers the entire bounding rectangle of the boundary geometry.
2. The graph's vertices are evenly spaced in the bounding rectangle given a target spacing. This even spacing produces a regular sampling of the space, which allows us to generalize our findings over the whole space. If not specified the spacing of the grid defaults to 1 m, which is a typical sample resolution used for simulating human behavior.[17]
3. The vertices are connected with edges to each of their 8-neighbors (the 8 other vertices directly adjacent to them). These edges form an even grid of straight and diagonal lines (Figure 1b).

Culling Edges

The next step is to remove any edges from the graph that intersect any of the polylines defining the boundary geometry (Figure 1c). The result of this operation is a disconnected graph with a separate subgraph in each enclosed space as defined by the boundary polylines. The computation requires (m × n) operations with m is the number of edges in the graph and n is the number of line segments defining the boundary objects. However, as the number of edges tends to be much higher than the number of boundary objects, the algorithm's complexity in effect scales linearly with the resolution of the analysis graph.

Connection of Sources and Sinks

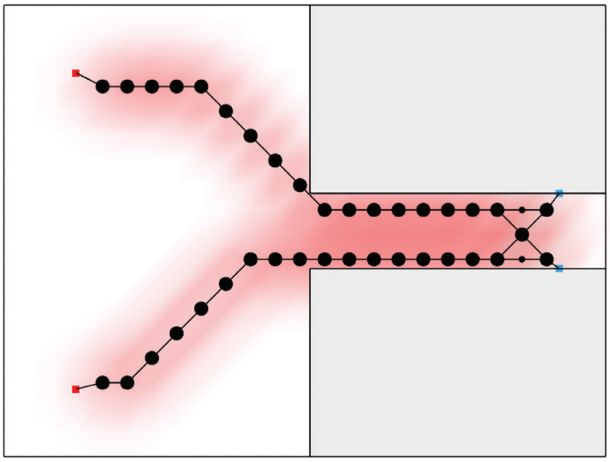
The next step is to integrate the source and sink points as vertices into the analysis graph by connecting each one to all of the vertices in the graph within a distance equal to the graph's target resolution. The closest point search necessary for this operation is sped up by first applying a k-d tree data structure to the vertices of the graph. As the source and sink points form new edges, they can reconnect sections of the graph that were separated in the previous operation. This can be useful when a source or sink point is a door, as the new edges will provide an access route between the subgraphs on either side of the door (Figure 1d).

Routing Paths

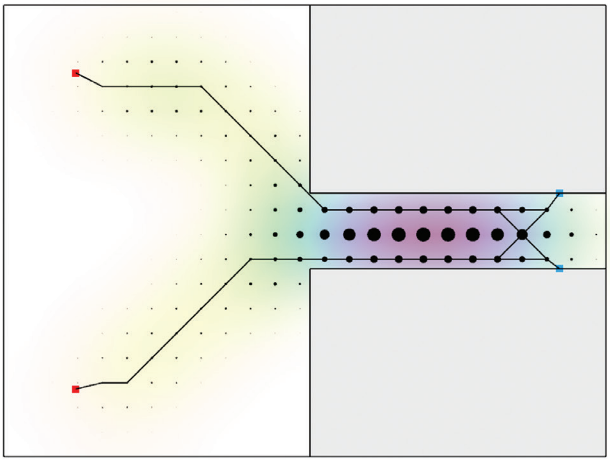
The next step is to route shortest paths through the analysis graph between each pair of source and sink points (Figure 1e). This routing is performed using the Dijkstra algorithm, which efficiently computes the shortest paths from a given vertex to every other vertex in a graph. As the routes are computed the vertices of the analysis graph store the total number of times they are traversed. The contribution of each path to the vertices' traversal counts can also be adjusted by specifying a weight to each pair of source and sink points. For example, a higher weight can be used to represent paths that generate more traffic, or a zero weight can be used to exclude nonpreferred paths.

Dissipation of Traversals

A major limitation of using graph-based shortest path calculations to simulate congestion in space is that the calculation only considers the shortest possible path and not the actual path that may be taken by a human occupant. This actual path is highly dependent on the nature of the space a person is walking through,[20] which is not



(a)



(b)

considered by the shortest path algorithm. For example, a group of people walking through a tight corridor may feel much more congested than the same group walking through a larger space. This is because in reality people will spread out more in a larger space to avoid other walkers instead of taking the absolute shortest route through the crowd. Meanwhile, the shortest path algorithm will easily route any number of people through the same sets of edges and predict the same level of congestion independent of the shape or quality of the space (Figure 2a).

Crowd simulation methods model this spreading out through dynamic behavioral simulations of individual human agents.[21] However, such a realistic simulation is often prohibitively slow and not necessary for schematic space planning. In our case, we simulate this spreading of traffic by "dissipating" the traversal data through the connected vertices of the analysis graph. At each time step of the dissipation process, each vertex splits its traversal value evenly among itself and its connected 8-neighbors (Figure 1f).

This dissipation operation gives us a more realistic prediction of congestion by smoothing out some of the particularities of the graph-based traversal calculation and dispersing the traversals through open areas of the floor plan. Because the graph is disconnected by the boundary geometry, vertices cannot dissipate their traversal data across walls or other boundaries. Thus, tight spaces with many traversals remain very congested, while more open spaces with the same number of traversals become relatively less congested (Figure 2b). Furthermore, as all the vertices are placed on an even grid, we can consider the dissipated traversal values as a low-resolution uniform sampling of congestion values across the space.

Although this is not a proven method and has not yet been validated with real-world observations, it is intuitive because it distributes congestion through the open areas of a plan similar to how people would disperse in an open space. It also matches our goals of not accurately simulating the dynamics of human movement and behavior but simply predicting the overall level of occupancy or congestion throughout the space.

Calculation of Metric

Based on the calculated traversal data in the graph, we compute the buzz metric to measure not only the amount of total congestion in the space, but the extent of its distribution throughout the space. First, we compute the centroid of all vertices, weighted by their dissipated traversal values. We then connect this centroid point as a vertex into the analysis graph and compute the shortest routes from each vertex with a traversal value higher than zero to the centroid.

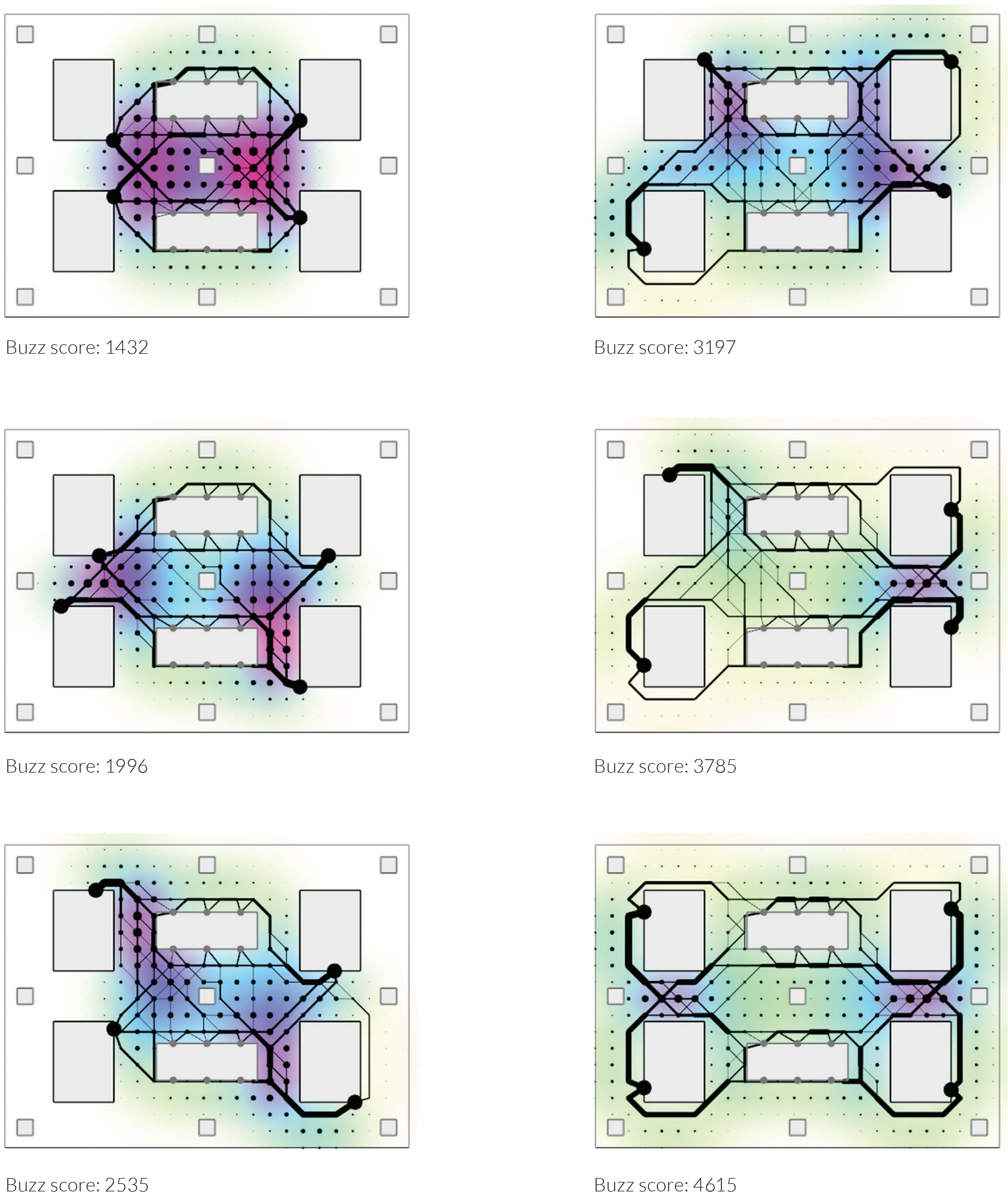
The buzz metric is the sum of the length of these routes, weighted by the traversal value of the starting vertex (Equation 1). This metric gives us a single-value measure of the distribution of high congestion areas in a way that considers the actual geometry and predicted occupancy of the space. The metric's value scales positively with both the total congestion in a plan and the average distance between zones of high congestion. The same amount of congestion clustered in one area will produce lower buzz scores than if it is spread between multiple areas throughout the plan.

$$buzz = \sum_{i=0}^{n} l_n * t_n$$

△ Equation 1. Calculation of buzz metric, where (n) is the number of vertices with more than one traversal, (l_n) is the length of the shortest route from vertex (**n**) to the computed weighted centroid, and (t_n) is the traversal value of vertex (**n**).

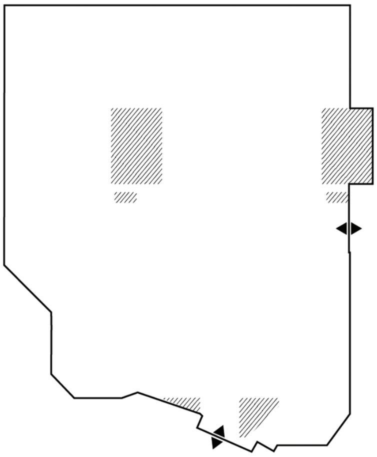
Figure 3 shows a demo using a simple office floor plan defined by two clusters of desks as sources of traffic and the doors of four shared meeting rooms as sinks. By moving the doors of the meeting rooms around the offices we can generate a range of buzz metric values. The results match our expectations that when doors are on the outside of the space, they will generate more productive congestion since the congestion will be dispersed into separate areas of the plan. However, if the doors are placed toward the interior work area the congestion will be too highly focused, likely creating distraction rather than buzz.

It is important to note that the goal of this method is not to

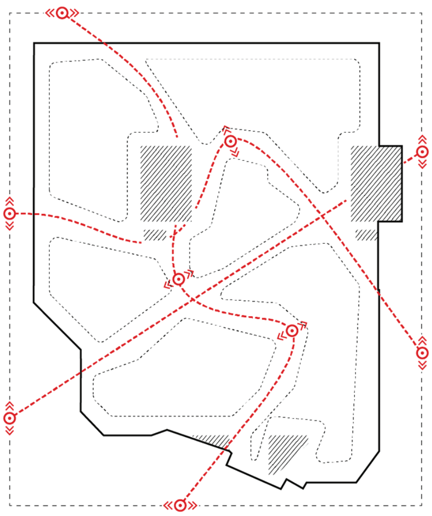


◁ Figure 2. Vertex traversals through a corridor space (a) before and (b) after dissipation, ©2017 The Living, an Autodesk Studio.

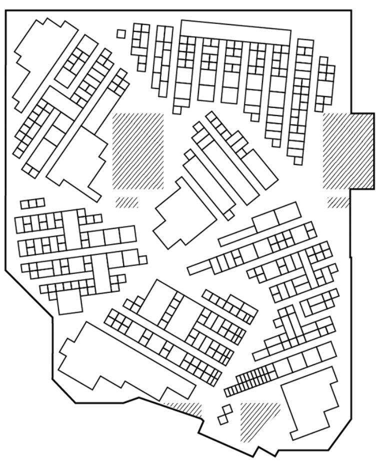
△ Figure 3. Comparison of floor layouts showing concentrations of congestion and associated buzz metric scores, ©2017 The Living, an Autodesk Studio.



(a) Input layout domains and main entrances



(b) Input avenues geometry



(c) Output layout geometry

produce a very accurate simulation of congestion or crowding behavior or to rival the accuracy of agent-based crowd simulation. Because our metric is calculated statically in a single time step it simply cannot capture the dynamic time-based behavior of actual people in space. Instead, the goal of the buzz metric is to provide a general idea of where concentrations of people may occur based on the shape of the space and the programs within it. Due to this abstraction, our method cannot be used for critical applications such as crowd control or evacuation simulations. However, since it captures the effect of both spatial layout and program to simulate overall occupation, it is sufficient for use during early-stage schematic design and space planning, especially if the design process involves an automated optimization workflow where many designs must be evaluated.

Case Study

To test the buzz metric on an actual application of generative design for space planning, we applied it to the design of a trade show in a large exhibit hall. This section will briefly describe the design problem, explain the geometry system used to generate different spatial layouts, and show how the buzz metric was used to evaluate the potential for productive congestion in each design option.

Problem Definition

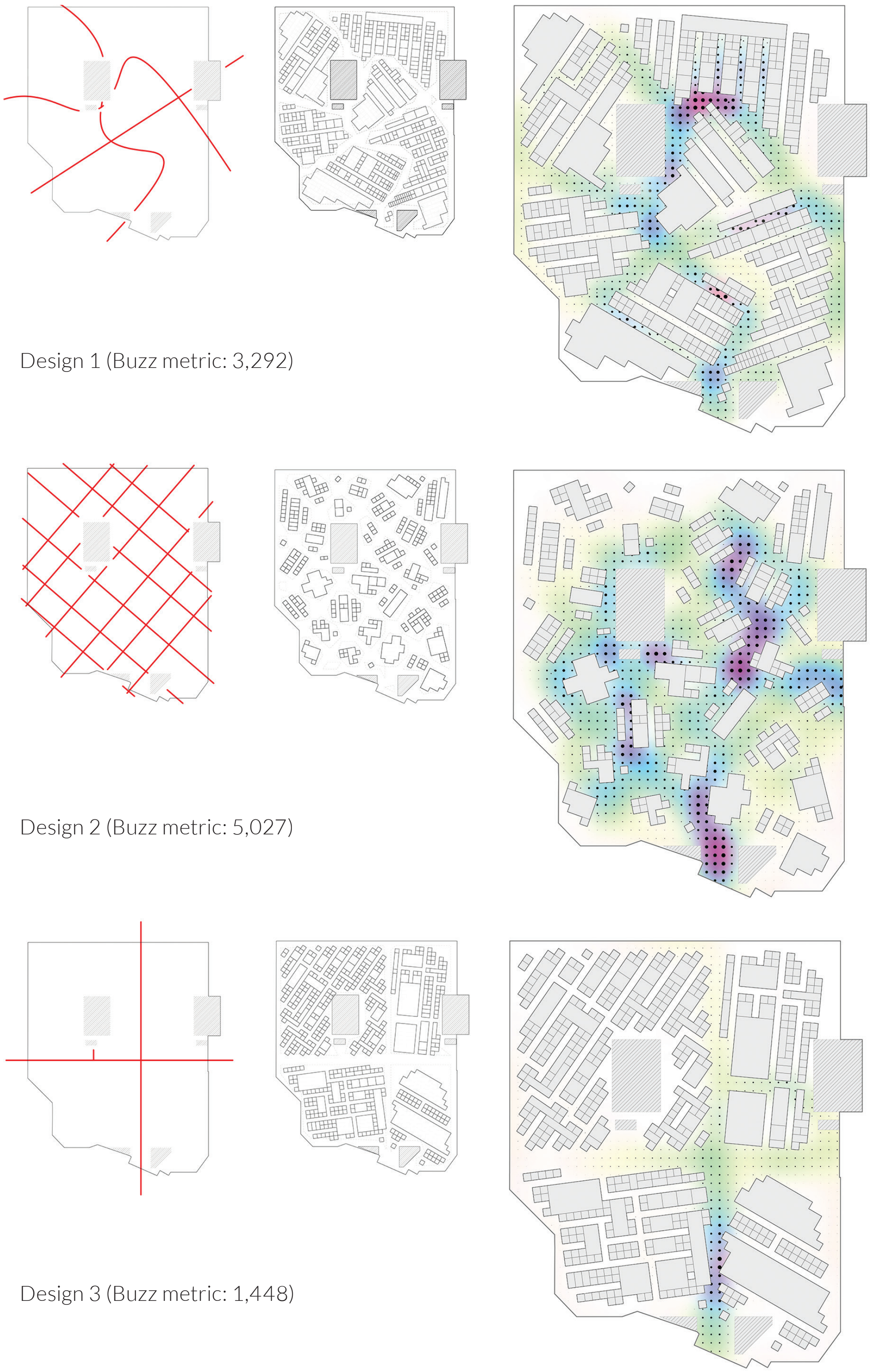
The design problem was to find the optimal layout of a specified number of booths and programs of various shapes and sizes within an existing exhibit hall. One of our main goals for the layout was to create a variety of distributed "buzz" areas where people would naturally gather and activate various programs in the space. To find the optimal design we used a generative design framework that could generate a large variety of plan layouts and evaluate them based on the buzz metric in order to find the optimal design.

Design Space Model

The first step was the construction of a parametric design space model that could generate a large variety of possible space layouts for the exhibit hall (Figure 4). Based on an urban development model, our system starts by laying out a series of "avenues" that subdivide the exhibit hall into a collection of smaller parcels. Major programs are then placed along the main avenues, and parcels are merged until the area requirements of all programs are met. Finally, the remaining parcels are populated with a collection of standard booth sizes.

Optimization

Once the design space is specified it can be explored through a metaheuristic optimization algorithm. In this case, we began the process with an exploratory even sampling of the entire design space using a sample of 20,480 designs. We then used a genetic algorithm based on NSGA-II[22] to run an optimization for 100 generations with 320 designs in each generation, requiring the evaluation of 32,000 additional designs. Thus, in total 52,480 designs needed to be individually evaluated until a final set of optimal designs was chosen. Each update of the model, including the calculation of the buzz metric, required 20 seconds on a standard



◁ Figure 4. Description of design space model, ©2017 The Living, an Autodesk Studio.

△ Figure 5. Geometry and buzz metric values for three designs of exhibition center, ©2017 The Living, an Autodesk Studio.

computer. As a result, the entire task was completed in under two weeks. Performing the same analysis would be impossible with a dynamic crowd simulation, since each design would take several hours to compute.

Observations

To see how the buzz metric performs in this model we can compare three plans, each having the same number and types of programs (Figure 5). The first design uses a mix of curved and straight avenues, the second uses an orthogonal grid, and the third uses a small set of larger straight avenues. As all plans have the same programs, the overall number of traversals is the same. Thus, the variation in buzz metric scores can be attributed to how these traversals are dispersed throughout the plan, with higher dispersal leading to higher buzz metric scores.

To evaluate whether the buzz metric matches our intuition, we can use a heat map to visualize the traversal values of the vertices in each plan. We can see that Design 3 scores the lowest on the buzz metric. This is likely due to the concentration of large programs and tight corridor near the lower entrance, which produces a pinch point leading to the concentration of congestion in a single area of the plan. In Design 1 the large programs are more distributed with more access routes from both entrances. This produces a wider distribution of congestion zones (as visualized by the heatmap) and a higher score for the buzz metric. In Design 2, the programs are highly distributed while the grid provides a range of possible pathways through the space. This produces the widest distribution of congestion in the space, leading to the highest buzz metric score.

Although these results match our intuition, they still need to be validated based on real-world data and observation. A post-occupancy analysis of the space could be performed to validate whether people concentrate in the zones of congestion predicted by the buzz metric. Ideally, two similar spaces could be constructed with different internal arrangements, and then visitors could be surveyed about which space they found to have more “productive” congestion. These survey results could then be compared to the single-value buzz metrics of the space to validate its predictions.

Conclusion

The novel method described in this paper offers a solution to the tradeoff between time and specificity seen in the current state-of-the-art in behavioral simulation of architectural space. Currently, designers who want to analyze the potential for productive congestion in a space must do so using a complex, dynamic, agent-based crowd simulation. However, these lengthy simulations are of minimal assistance to those who want to incorporate generative design into their space planning process, which often relies on the evaluation of hundreds if not thousands or more design options until a suitable design is found. This method gives designers a way to simulate the potential for productive congestion in a space based on actual predicted human behavior but calculated in a static way that can be directly integrated into an automated generative design workflow. Such a method can be extremely useful to designers who want to capitalize on the potentials of generative design to create better, more active, and higher performing public spaces.

In addition to validating our model as previously described, there are several subsequent steps for the development of this methodology. First, an extension of the method to target the creation of productive congestion in specific areas of the floorplan or tied to specific programs would be beneficial. Currently, the metric only measures the overall distribution of congestion, which takes into account the shape of the space and the predicted movement of people, but does not give preference to any particular location where the congestion should occur. Furthermore, additional investigations into improving the efficiencies of the algorithm and scaling it up to a much larger area such as an entire shopping mall or a multi-floor office building would be beneficial.

Finally, additional research should integrate more specific human behavioral aspects into the model to achieve even more accurate results. Many state-of-the-art crowd simulation systems benefit from using a variety of behavioral profiles for the agents in their simulations.[23] Similar differences in behavior can be integrated by specifying variable parameters to each route that can change the way a route is calculated and how it contributes to the overall calculation of congestion in the space. By integrating more detailed behavioral models into the method, higher accuracies in the simulation can be achieved while maintaining the speed of a static computational method.

Notes

1. Greg Lindsay, "Engineering Serendipity," *New York Times* (2013), http://www.nytimes.com/2013/04/07/opinion/sunday/engineering-serendipity.html.
2. Rachel Emma Silverman, "The Science of Serendipity in the Workplace," *Wall Street Journal* (2013): B6.
3. Chloë Brown, Christos Efstratiou, Ilias Leontiadis, Daniele Quercia, and Cecilia Mascolo, "Tracking Serendipitous Interactions: How Individual Cultures Shape the Office," in *Proceedings of the 17th ACM conference on Computer Supported Cooperative Work & Social Computing* (New York: ACM, 2014), 1072–1081.
4. Alex Pentland, "The New Science of Building Great Teams," *Harvard Business Review* 90, no. 4 (2012): 60–69.
5. Barry P. Haynes, "The Impact of Office Comfort on Productivity," *Journal of Facilities Management* 6, no. 1 (2008): 37–51.
6. Steve Whittaker, David Frohlich, and Owen Daly-Jones, "Informal Workplace Communication: What Is It Like and How Might We Support It?," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, edited by Beth Adelson, Susan Dumais, et al. (New York: ACM, 1994), 131–137.
7. Ben Waber, Jennifer Magnolfi, and Greg Lindsay, "Workspaces That Move People," *Harvard Business Review* 92, no. 10 (2014): 68–77.
8. Steve Gwynne, E. R. Galea, M. Owen, Peter J. Lawrence, and L. Filippidis, "A Review of the Methodologies Used in the Computer Simulation of Evacuation from the Built Environment," *Building and Environment* 34, no. 6 (1999): 741–749; Jérôme Comptdaer, Emmanuel Chiva, Stéphane Delorme, Henri Morlaye, Jérôme Volpoët, and Olivier Balet, "Multi-scale Behavioral Models for Urban Crisis Training Simulation," in *Proceedings of BRIMS, 16th Conference on Behavior Representation in Modeling and Simulation, Norfolk*. 2007: (Red Hook, NY: SISO, 2009), http://www.proceedings.com/05592.html.

9. George Drettakis, Maria Roussou, Alex Reche, and Nicolas Tsingos. "Design and Evaluation of a Real-world Virtual Environment for Architecture and Urban Planning." *Presence: Teleoperators and Virtual Environments* 16, no. 3 (2007): 318–332.

10. Stephanie Huerre, Jehee Lee, Ming Lin, and Carol O'Sullivan. "Simulating Believable Crowd and Group Behaviors," in *ACM SIGGRAPH ASIA 2010 Courses* (New York: ACM, 2010), 13, http://dl.acm.org/citation.cfm?id=1900533.

11. Xia Li, and Xiaoping Liu, "Embedding Sustainable Development Strategies in Agent-based Models for Use as a Planning Tool," *International Journal of Geographical Information Science* 22, no. 1 (2008): 21–45.

12. Gideon Aschwanden, Simon Haegler, Frédéric Bosché, Luc Van Gool, and Gerhard Schmitt, "Empiric Design Evaluation in Urban Planning," *Automation in Construction* 20, no. 3 (2011): 299–310.

13. Tian Feng, Lap-Fai Yu, Sai-Kit Yeung, K. Yin, and Kun Zhou, "Crowd-driven Mid-scale Layout Design," *ACM Transactions on Graphics* 35, no. 4 (2016): 1-14.

14. Ibid.

15. R. Ratti, "Urban Texture and Space Syntax: Some Inconsistencies," *Environment and Planning B: Planning and Design* 31, no. 3 (2004): 487-499.

16. B. Hanson Hillier and Julienne Hanson, *1984: The Social Logic of Space* (Cambridge: Cambridge University Press, 1993); ; Alasdair Turner, Alan Penn, and Bill Hillier, "An Algorithmic Definition of the Axial Map", Environment and Planning B: Planning and Design, 32, no. 3 (2005): 425-444.

17. Alasdair Turner, Maria Doxa, David O'Sullivan, and Alan Penn, "From Isovists to Visibility Graphs: a Methodology for the Analysis of Architectural Space", *Environment and Planning B: Planning and Design*, 28, no. 1 (2001): 103-121.

18. Alan Penn and Alasdair Turner, "Space Syntax Based Agent Simulation", Bartlett School of Graduate Studies, London. Presented at: 1st International Conference on Pedestrian and Evacuation Dynamics, University of Duisburg, Germany (2001).

19. Jay Stucky and Dan Lee, "On Applying Viewshed Analysis for Determining Least-cost Paths on Digital Elevation Models," *International Journal of Geographical Information Science* 12, no. 8 (1998): 891–905.

20. Julienne Hanson, "Decoding Homes and Houses", *Environment and Planning B: Planning and Design*, 21, no. 6 (1994): 675-704.

21. Olfa Beltaief, Sameh El Hadouaj, and Khaled Ghedira, "Multi-agent Simulation Model of Pedestrians Crowd Based on Psychological Theories," in *Logistics (LOGISTIQUA), 2011 4th International Conference on* (IEEE, 2011), 150–156.

22. Deb Kalyanmoy, Amrit Pratap, Sameer Agarwal, and T. A. M. T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation* 6, no. 2 (2002): 182–197.

23. Zhendan Xue and Christina Bloebaum, "Human Behavior Effects on Cabin Configuration Design Using VacateAir," in *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition, Orlando* (2009), 42, https://arc.aiaa.org/doi/abs/10.2514/6.2009-42.

Danil Nagy is a Principal Research Scientist at The Living, an Autodesk Studio. His work and research focuses on computational design, generative geometry, advanced fabrication, machine learning, and data visualization. Danil has spoken and presented work at a variety of conferences and venues, including the Design Modelling Symposium, Biofabricate, Techonomy:Bio, and ACM SIGGRAPH.

Lorenzo Villaggi is an Associate Research Scientist at The Living, an Autodesk Studio. His work focuses on generative design, new materials, and novel forms of visualization. Lorenzo also co-founded and co-edits: (pronounced "colon"), a collective workshop on architectural practices based in New York City. His work has been exhibited in many internationally renowned venues including Milan Design Week and the New Museum

Jim Stoddart is an Associate Research Scientist at The Living, an Autodesk Studio. His work focuses on applications of novel technologies to real-world design problems, including new materials, development of custom digital fabrication workflows, and exploration of new visualization technologies. His work has been exhibited in a variety of venues including the Chicago Architecture Biennial and the Museum of Modern Art.

David Benjamin is Founding Principal of The Living, an Autodesk Studio. David has lectured about his work in many parts of the world, and he currently teaches at Columbia Graduate School of Architecture, Planning and Preservation. Before receiving a Master of Architecture from Columbia, he received a Bachelor of Arts from Harvard.

PEER REVIEW / SIMULATIONS