Design Loop: Calibration of a Simulation of Productive Congestion Through Real-World Data for Generative Design Frameworks

Lorenzo Villaggi[1(✉)], James Stoddart[1], Pan Zhang[2], Alex Tessier[2], and David Benjamin[1]

[1] The Living, an Autodesk Studio, New York, NY, USA
lorenzo.villaggi@autodesk.com
[2] Autodesk Research, Toronto, Canada

Abstract. This paper extends the applicability of generative design for space planning frameworks for ongoing and guided post-occupancy modifications. It involves the comparison of a graph-based productive-congestion simulation with empirical data and the use of a metaheuristic search algorithm to calibrate and fine-tune simulation parameters for greater accuracy. This methodology is demonstrated through a real-world generative designed case-study and the post-occupancy collection and processing of movement data through custom computer vision workflows.

Keywords: Simulation calibration · Post-occupancy · Generative design

1 Introduction

Generative design (GD) leverages the power of computation to explore large design spaces and discover novel and high-performing solutions relative to a set of goals and constraints. This process relies on parametric modeling software to represent large solution spaces, simulation software to evaluate each generated design, and metaheuristic optimization solvers, such as genetic algorithms, to search the design space for optimal solutions [1]. This paper showcases how to use empirical occupancy data to automatically tune simulation parameters, minimize assumptions used in the initial built design iteration, and offer a concrete methodology for planners and designers to create more accurate results for future changes for similar design problems (Fig. 1).

1.1 Designing for Uncertainty

Typical design processes tend to focus on satisfying immediate or "known" requirements. Any misalignment between built architectural features and occupant needs—by fault of either incorrect assumptions or the inevitable evolution of individual preferences—are deferred as a separate future design problem. Likewise, a tight focus on addressing immediate problems without considering future needs risks making designers "unable to predict the longer-term consequences in use of what we design" [2]. Broad adoption of these attitudes produces detrimental consequences including

C. Gengnagel et al. (Eds.): DMSB 2019, *Impact: Design With All Senses*, pp. 376–389, 2020.
https://doi.org/10.1007/978-3-030-29829-6_30

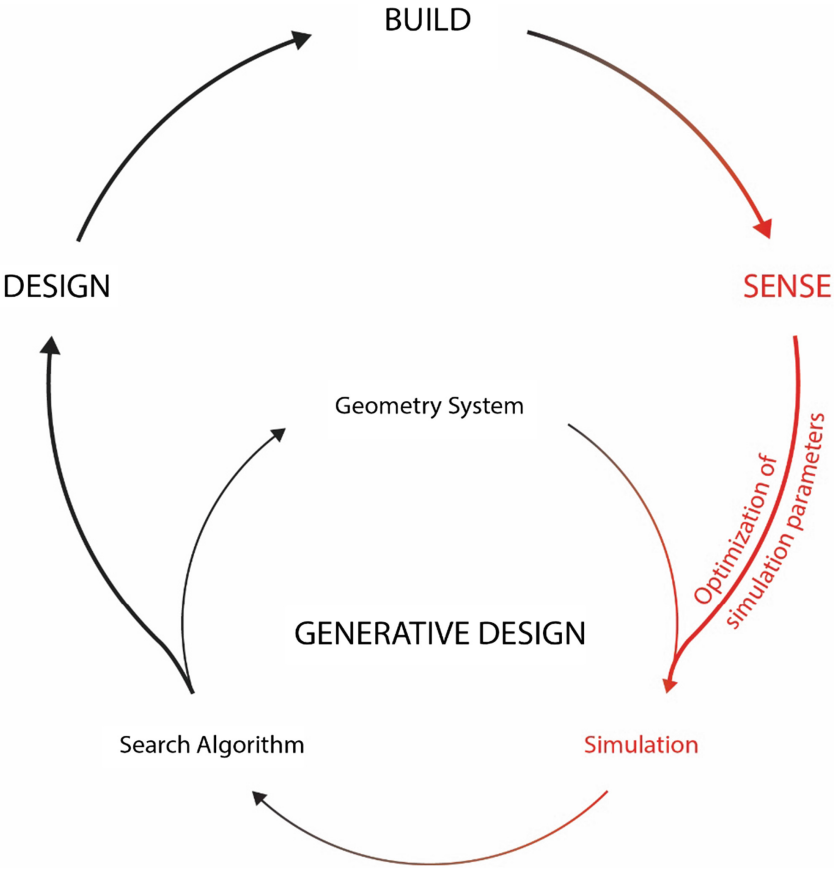

Fig. 1. Design loop: a framework for tuning generative design models to meet new building requirements through empirical data and hyperparameter optimization methods.

incompatibility with user needs, low space utilization leading to high-turnover of tenants, increased operational and maintenance costs, and poor energy efficiency. "More than any other human artifact, buildings excel at improving with time, if they are given a chance" [3]. In this spirit, this paper demonstrates a flexible generative design framework that over time can semi-automatically learn about how space is used and reduce initial assumptions to make data-informed decisions about future changes.

1.2 Productive Congestion

Productive congestion is a critical occupant-level spatial quality that is gaining interest within interior architecture and space planning as a measure of the degree of liveliness and activation of interior spaces [4–6]. Despite the large literature on the benefits of curated congestion, there are no replicable methods to design spaces that encourage such phenomena let alone evaluate it. To address this, our group has previously

demonstrated a general-purpose static simulation technique that quantifies such behavior and is compatible with generative design workflows [7]. The work also involved the application of generative design for the design of a mid-scale exhibit hall for 7000 visitors over a three-day conference and trade-show event [8, 9]. The goal was to configure a layout of booths and pavilions that maximized the spread and intensity of high activity zones across the hall and increased the level of exposure of each program to high traffic areas (Fig. 2). This paper furthers this work by offering a generalizable framework that calibrates the simulation parameters through observed congestion data collected from the built layout to improve its fidelity to real-world behavior, thus improving the overall results of the generative design model.

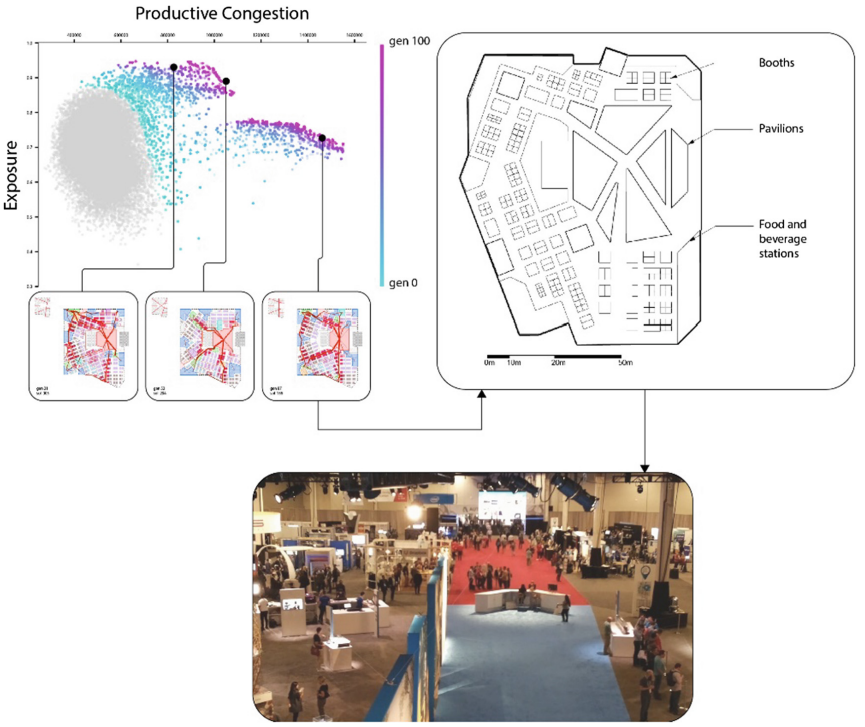

Fig. 2. The optimization dataset derived from applying a framework for generative design for architectural space planning (*top*) was used to design the final built exhibit hall layout (*right and bottom*). The process involved the simulation of productive congestion to automatically evaluate the amount of high activity zones on each generated layout.

2 Related Work

Discrepancies between simulated results and real-world phenomena are common issues in building simulations [10, 11]. Since more of these models are actively used in decision-making processes, the study of calibration methodologies—the process of

adjusting simulation parameters to match target results based upon empirical observation data [12] —is of growing interest. Calibration methods can be grouped into two main areas: manual parameter adjustments and semi to fully automatic parameter tuning strategies [13]. Due to the complexity of our model and the number of input parameters required to be calibrated we implemented an automated simulation tuning framework. In this regard, automatic calibration techniques mostly revolve around deterministic search methods like gradient-based ones and linear regression [14]. While these may yield effective results in a short amount of time, they tend to be limited by local optima [12] which is why our method relies on a metaheuristic stochastic search (GA) that is also able to deal with highly discontinuous design spaces [15]. Recent work demonstrated the use of evolutionary algorithms (EA) for calibration of crowd simulation models [16] but while this focused on accurate microscale behavior, our work privileges predictions of distributions at larger scales of traffic density which are faster to compute thus more compatible with generative design frameworks.

3 Methodology

This section describes the details of our workflow which involves the incorporation of the productive congestion simulation into GD for the design of a real-world case study (Fig. 2), the collection of congestion data via video footage, the processing of such data to be comparable with our simulation results and an automated calibration of the simulation parameters to minimize the difference between simulated results and observed data.

3.1 Simulation Assumptions

As our method to measure productive congestion has been described in detail in past work from our group [7] we will limit the following description to the key assumptions and summary of techniques that our model utilizes. Given any 2D floor plan represented as graph data structure, the simulation generates a network of walkable paths in the form of a set of nodes and edges and physically routes simulated occupants in space based on pairs of a source (point of origin) and a sink (destination). As the routes are computed, each node in the graph stores an aggregate count of the number of times it has been traversed (traffic). To create more realistic results, the computed traffic at each node is iteratively diffused across its adjacent nodes with flow-blocking across defined geometric barriers. This accommodates for the natural variation of human pathfinding and intensifies congestion in spaces with physically confining boundaries and dilutes it in more open spaces (Fig. 3).

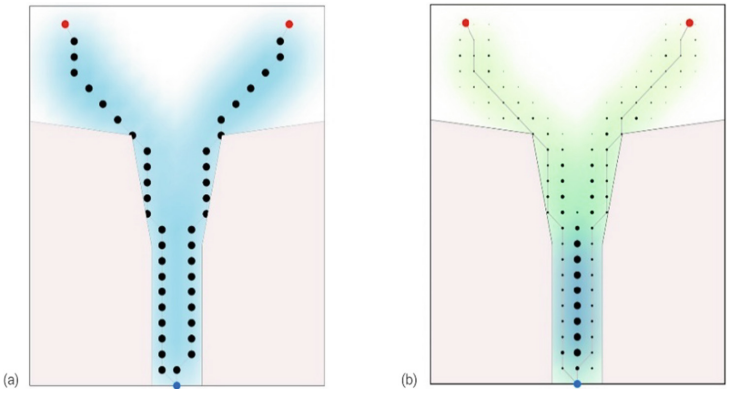

Fig. 3. Typical shortest path-based congestion results are independent of the quality of space (a), while our technique takes into consideration amount of people and geometrical context (b)

For each source and sink node the computed traversals can be adjusted through the manipulation of normalized weights (Fig. 4d). Each weight represents the level of desirability of the associated areas and are generally set on the basis of intuition and past experience.

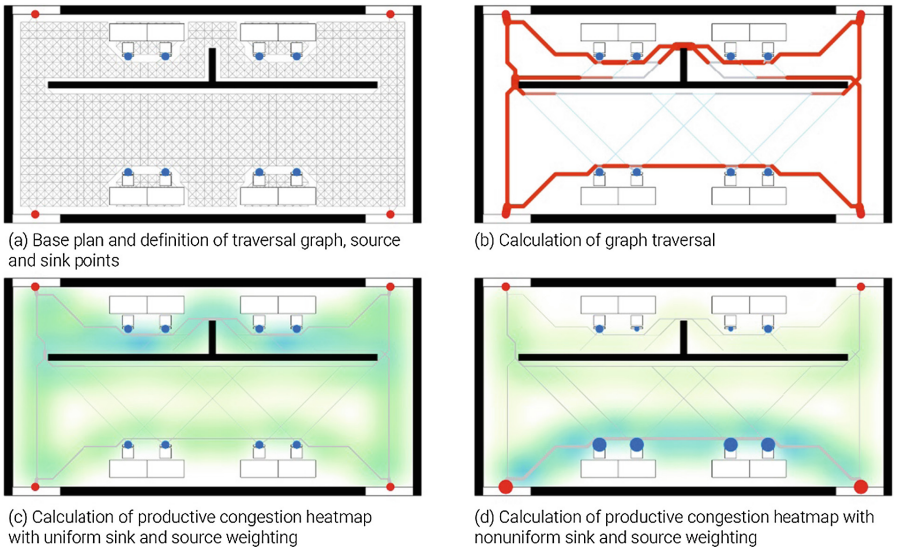

Fig. 4. Construction of traversal graphs and computation of productive congestion heatmap and weighting setting comparison.

To design the exhibit hall the weights were split into programmatic categories and manually assigned by all stakeholders involved. Values of 1.0 were assigned to the main pavilions and access and exit doors, which were assumed, from past year events, to generate high traffic. Similarly, 0.75 was assigned to food and beverage stations and other areas while 0.5 was assigned to all sponsor booths (see Table 1). Such assumptions can be reduced through an expanded framework of generative design that incorporates observed data to improve the accuracy of its results and drive future post-occupancy modifications.

Table 1. List of programs placed in the designed layout along with the manually assigned weights based on past experience and intuition.

Programs	Weights [0.00, 1.00]	Programs	Weights [0.00, 1.00]
Main pavilion 0	1.00	Store	1.00
Main pavilion 1	1.00	Food and beverage	1.00
Main pavilion 2	1.00	Industry booth 0	0.50
Main pavilion 3	0.75	Industry booth 1	0.50
Main pavilion 4	1.00	Industry booth 2	0.50
Hardware booth	0.25	Industry booth 3	0.50
Info booth	0.50	South access/exit	0.75
Drone booth	1.00	East access/exit 1	1.00
Software booth	0.75	East access/exit 2	1.00

3.2 Measuring Congestion Through Computer Vision

To validate the metrics used to generate the design, we outfitted the final built design to monitor usage and traffic flow in the post occupancy phase via video. The video data was collected and processed through a custom computer vision (CV) pipeline to extract human movement and generate aggregate counts and distribution heatmaps. Sensing with video offers many benefits including the density of information that can be extracted and offering a level of coverage and localization not feasible with manual counting methods, which, as discussed in [17], can greatly vary. However, use of video also required considerations to mitigate ethical and privacy concerns. In our case, all event participants had given prior consent to recording, and our methodology explicitly avoided any techniques that could be used to track individuals like facial or gait recognition.

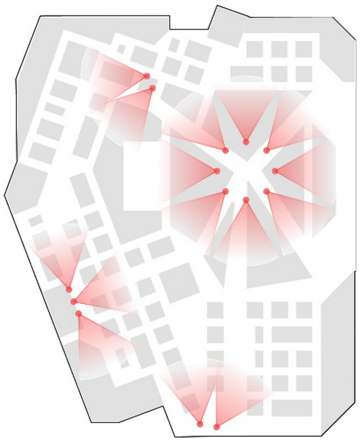

Fig. 5. Cameras setup and field of view coverage

The setup used 16 Raspberry Pi 3 (RPi) with RPi RGB cameras recording 1080p video positioned at key areas of the exhibit hall (Fig. 5). The RPi camera configuration was chosen for its low cost and high degree of flexibility from a software and networking perspective. The RPis were networked via Ethernet to a central server using Simple Network Time Protocol (SNTP) to synchronize timestamps. We also implemented a circular frame buffer which used simple change detection algorithms to limit saved frames to motion-events longer than 1 s.

All frames were batch processed post-event using 3D skeletonization [18] and homography [19] to create a transformation matrix mapping participant footsteps and location in 3D space. Skeletonization was captured using the OpenPose library [18], which generates multi-person keypoint detection and outputs a JSON file containing skeleton points of a recognized human form and a prediction confidence score (Fig. 6a). This data is generated on a per-frame basis with no frame-to-frame object tracking matching a generated skeleton to subsequent frames in a stable manner. For our purposes, we implemented a custom tracking system on top of the results from OpenPose.

Our tracking method uses the head of a person as our focal tracking object, as it is relatively stable in relation to the movement of the person and prominently visible from our downward facing camera setup. We used a confidence threshold to limit tracking to only high-confidence detections resulting in variable gaps in frames between detections. To compensate, we must decide if the head detected in frame n is the same head detected in prior frame $n - i$ where $i > 1$. This was accomplished using a distance threshold function which evaluates a head u in frame $n - i$ and a head v in frame n. Because there are many heads tracked simultaneously, we record the position of head v in the frame n as h_v^n in pixel coordinates. These positions are then used to compare, over the course of i frames, the positional deviation between heads u and v to

see if it falls below the distance threshold T and therefore likely the same individual, using this test function:

$$\left|\left|h_u^{n-i} - h_v^n\right|\right| < T\left(i, h_u^{n-i}\right) \tag{1}$$

T is a pixel distance in the image calculated empirically and tuned based on framerate and camera intrinsics. In our case, $T\left(i, h_u^{n-1}\right) = i \times D\left(h_u^{n-1}\right)$ where $D\left(h_u^{n-1}\right)$ is the average distance measured in pixels between two frames.

From the tracked individuals, we extracted the associated foot keypoints (Fig. 6b). Standard homographic methods from the OpenCV library [19] and known camera locations were used to project the 2D pixel positions in each frame to the floorplan. These projected movement paths were then counted for the three days of the event allowing for various levels of aggregation and analysis (Figs. 6c and d). To match the simulated data, the global aggregate sum was calculated across the covered areas.

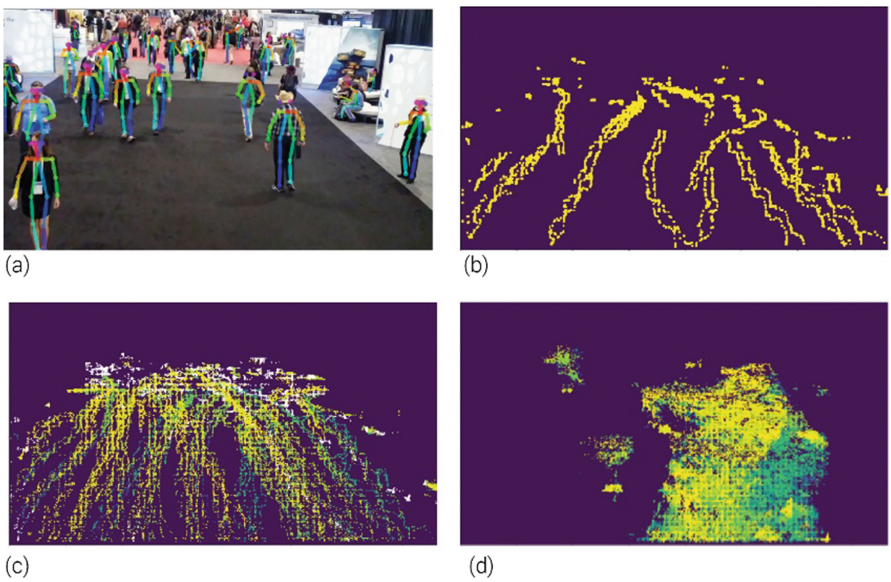

Fig. 6. Pedestrian path tracking from video recordings of the exhibit hall activity through computer vision pipelines. (7a) skeletonization; (7b) skeleton paths; (7c) tracking; (7d) aggregation over time.

3.3 Data Processing

To ensure parity across simulated and observed data, raw counts extracted from the video were upsampled and discretized to match the resolution of the computed simulation graph using bilinear interpolation. This method was chosen to maintain the fidelity of source data including its incomplete coverage. The remapped data samples were then normalized by aggregate counts over the duration of the data recording period to match the domain of simulated results (Fig. 7).

Fig. 7. Productive congestion visualization samples: (a) sample heatmap of productive congestion overlaid on exhibit hall plan layout. Blue and red dots represent respectively sink and source points. Their size reflects assigned weights; (b) heatmap of observed congestion data; (c) Euclidean difference between simulated and observed data; (d) dashboard showing weights grouped by program: of each rectangle, the width reflects assigned weight value, while the height reflects program surface area.

3.4 Calibration

To improve the accuracy of simulation results we setup an optimization framework that stochastically searches for different combination of weight parameters to minimize the difference or distance (fitness function) between simulated and observed data. With many optimization problems, this distance can be measured arithmetically using Euclidean or other basic distance methods, however our simulation assumes a constant quantity of gross traffic meaning the global average value of any configuration would be equal and thus the arithmetic distance between solutions would be negligible. Additionally, global arithmetic distance penalizes inexact results, and for our purposes, a simulated peak traffic zone that is only slightly misaligned with real-world observations is still considered "good". Some disagreement is expected between real-world results and simulation, as our graph-based method operates on an approximate model of movement that enables fast computation but doesn't capture the full complexity and inherently problematic nature of human behavior. As such, our optimization requires a fitness function that evaluates proximity to an ideal rather than precision.

Our method employs Wasserstein distance, a statistical distance method more commonly referred to as earth mover's distance (EMD), which evaluates the difference in probability distributions by measuring the "cost" as a function of quantity and distance of energy that must be relocated to reach a target distribution [20]. This relocation is evaluated locally meaning a slight misalignment between observed and simulated results with otherwise correct intensity distributions would have low impact on the fitness score. This metric is also compatible with incomplete real-world data as an input, as the deviation and transport costs are optimally computed between local distributions.

Our fitness model evaluates the distance between clusters of people from our simulated model and real-world results, represented by a probability distribution P_r. Our simulation model is denoted as a function, $S(L, \theta)$, where θ represents all model parameters and L is the given design layout. The probability distribution of our model $P_r(S(L, \theta))$ can be expressed as the sum of the distribution of all sub-clusters over the space, and where w_i, the distribution of a sub-cluster parameterized by a point, is measured using simple Gaussian sampling.

$$P_r(S(L, \theta)) = w_1 + w_2 + \ldots + w_i + \ldots + w_n \tag{2}$$

In addition to our simulated model flow of people $S(L, \theta_s)$, we can also represent the observed real flow of people using the same model with unique parameters $S(L, \theta_o)$. Thus, the objective function can be expressed as:

$$\Omega = ||P_r(S(L, \theta_s)) - P_r(S(L, \theta_o))||_w \tag{3}$$

where $||.||_w$ is the EMD. We also note that Ω is at its minimum when θ_o is equal to θ_s. Because Ω is not convex and may lack a single optimal solution, the use of a meta-heuristic search algorithm is well suited to this problem.

Our optimization used a variant of the NSGA-II genetic algorithm [21] for single objective search to minimize the fitness function Ω by controlling the input weights assigned to programmatic categories with the following optimization parameters:

- Population size per generation: 52
- Number of generations: 50
- Mutation rate: 0.3
- Crossover rate: 0.95

The tuning of optimization parameters is experiment-based and highly dependent on the problem to solve [22]. While a comprehensive comparative set of parameter tuning tests goes beyond the scope of this paper, we performed a few experiments based on past experience of similar single-objective problems and identified the set of parameters that returned good enough results for this particular problem.

4 Results

Through optimization, we saw a global improvement and convergence between simulated and real-world results with a relative reduction of EMD by 50% (Fig. 8). This indicates that the updated simulation parameters were able to reduce both the distance between zones of movement intensity and the difference in relative intensity from the empirical measurements in the areas where they were gathered.

Visualizing the difference allows for a nuanced evaluation of gains from our original simulation weighting. Improvements are seen in key layout areas, including the entrance and central focal spaces. However, the areas of data capture coverage tend to align with these key areas - broader coverage and more extensive sampling in spaces with a wider range of predicted traffic would allow for better validation. Likewise, the assumed point-to-point travel of the source metric is incapable of recreating some of the more diffused activity seen in the real-world data, such as the space across the entrance. Manipulations to the placement and quantity of sources and sinks and the protocol for generating paths could potentially provide distributions more aligned with the observed.

Because our input parameters represent comprehendible values (desirability of a given program) it is useful to examine the deviation between our initial assumptions for these weights and the values derived by the optimization process. Here we see deviations roughly correlating with areas of real-world data coverage, with programs in the tracked spacing maintaining predicted distributions and programs in non-captured areas seeing more drastic deviations (e.g. the food and beverage stations which saw a drop of 89%). This most likely represents an overfitting of the model, which would require adjustments to the fitness function to evaluate distance from the initial predicted weights in areas unseen by empirical data.

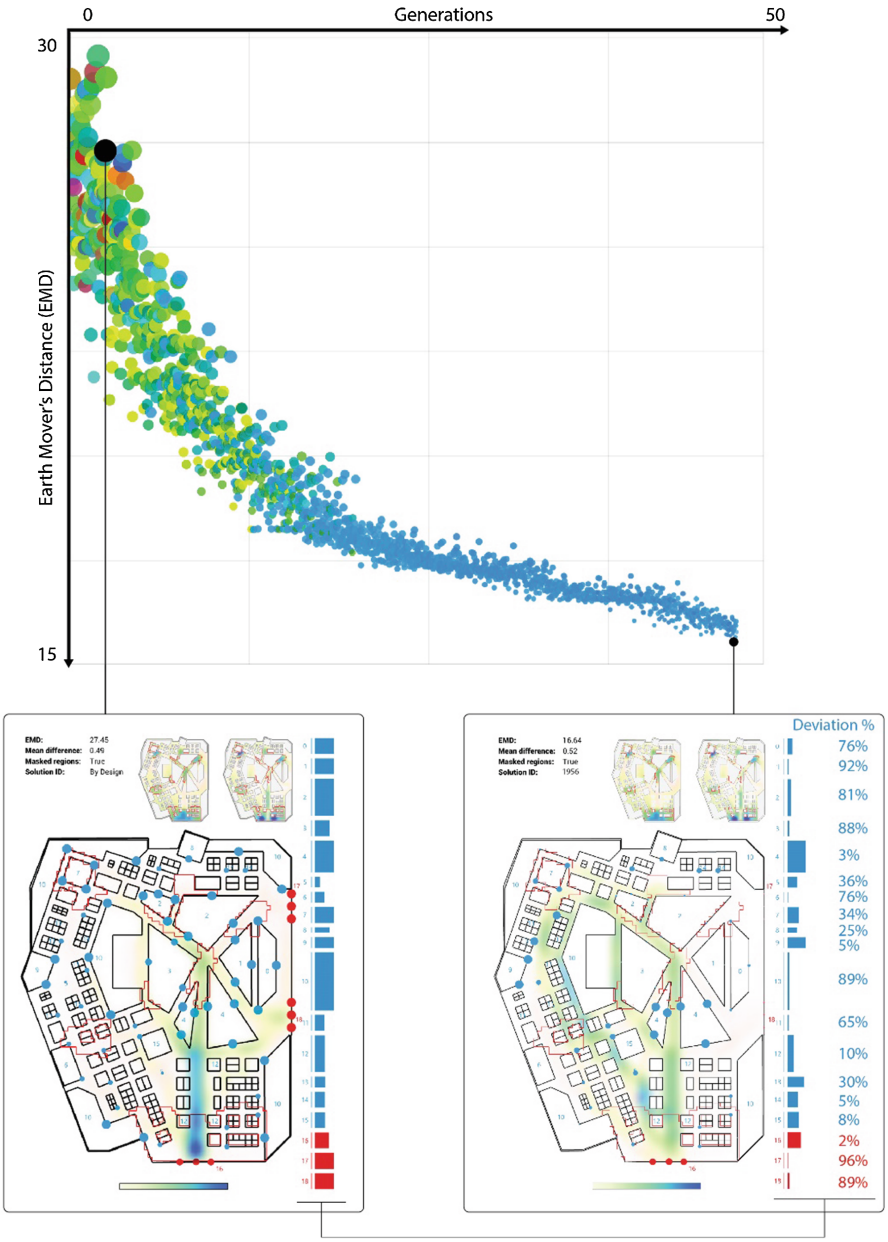

Fig. 8. Scatterplot of optimization data of simulation parameters (*top*) color-coded by input-space: consistent color in younger generations reflects optimization convergence; simulation results used for the design of the layout (*bottom left*) (mapped in the scatterplot for benchmarking purposes); solution with minimal EMD and calibrated weights with deviation data (*bottom right*).

5 Conclusions and Next Steps

Our method demonstrates the viability of using post-occupancy data to validate and calibrate the behavior of metric models used in a generative design process. Simulation of unpredictable behavior is itself a design problem, and this approach balances the intuition of architectural training with the evidence-based rigor of data-driven analysis. These methods allow constructive critical analysis of design methods and assumptions, while enabling greater confidence in design decision making powered by data-informed feedback loops.

This example also highlights the need for further exploration, including the generation of new designs with the updated metric to validate the difference between resultant designs from the old model and the new. Likewise, it provides comparative feedback for refinement of the simulation techniques including where approximate behavior deviates and matches observed patterns.

Acknowledgments. We thank Liviu Calin and John Yee from Autodesk Research for leading and assisting with the development of the tracking pipeline and its implementation.

References

1. Nagy, D., Lau, D., Locke, J., Stoddart, J., Villaggi, L., Wang, R., Zhao, D., 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, p. 7. Society for Computer Simulation International (2017)
2. Duffy, F.: Work and the City. Edge Futures/Black Dog Publishing, London (2008)
3. Brand, S.: How Buildings Lear: What Happens After They're Built. Penguin Books, New York (1994)
4. Lindsay, G.: Engineering serendipity. New York Times. https://www.nytimes.com/2013/04/07/opinion/sunday/engineering-serendipity.html. Accessed 09 June 2019
5. Silverman, R.E.: The science of serendipity in the workplace. https://www.wsj.com/articles/SB10001424127887323798104578455081218505870. Accessed 09 June 2019
6. Brown, C., Efstratiou, C., Leontiadis, I., Quercia, D., Mascolo, C.: Tracking serendipitous interactions: how individual cultures shape the office. In: Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing, pp. 1072–1081. ACM (2014)
7. Nagy, D., Villaggi, L., Stoddart J., Benjamin, D.: The Buzz Metric: a graph-based method for quantifying productive congestion in generative space planning for architecture. Technol.|Arch. + Des. **1**, 64–73 (2017)
8. Villaggi, L., Nagy, D.: Generative design for architectural space planning: The Case of Autodesk University 2017 Layout. https://www.autodesk.com/autodesk-university/class/Generative-Design-Architectural-Space-Planning-Case-Autodesk-University-2017-Layout-2017. Accessed 08 June 2019
9. Nagy, D., Villaggi, L., Stoddart, J., Benjamin, D.: Beyond heuristics: a novel design space model for generative space planning in architecture. In: ACADIA Conference Proceeding, pp. 436–445 (2017)

10. Dahabreh, I., Issa, J., Chan, J.A., et al.: A review of validation and calibration methods for health care modeling and simulation. In: Modeling and Simulation in the Context of Health Technology Assessment: Review of Existing Guidance, Future Research Needs, and Validity Assessment, https://www.ncbi.nlm.nih.gov/books/NBK424022/. Accessed 31 Mar 2019
11. Oreskes, N., Shrader-Frechette, K., Belitz, K.: Verification, validation, and confirmation of numerical models in the earth sciences. Science **263**(5147), 641–646 (1994)
12. Yuan, J., Hui Ng, S., Leung Tsui, K.: Calibration of stochastic computer models using stochastic approximation methods. IEEE Trans. Autom. Sci. Eng. **10**(1), 171–186 (2013)
13. Edwards, R.E.: Automating large-scale simulation calibration to real-world sensor data. Ph. D. dissertation, University of Tennessee (2013)
14. Zhong, J., Hu, N., Cai, W., Lees, M., Luo, L.: Density-based evolutionary framework for crowd model calibration. J. Comput. Sci. **6**, 11–22 (2015)
15. Wolinski, D., Guy, S.J., Olivier, A.H., Lin, M., Manocha, D., Pettré, J.: Parameter estimation and comparative evaluation of crowd simulations. Comput. Graph. Forum **33**(2), 303–312 (2014)
16. Johansson, A., Helbing, D., Shulka, P.K.: Specification of a microscopic pedestrian model by evolutionary adjustment to video tracking data. In: Advances in Complex System, vol. 25. World Scientific Publishing Company (2008)
17. Goodier, R.: The curious science of counting a crowd. Popular Mechanic. https://www.popularmechanics.com/science/a7121/the-curious-science-of-counting-a-crowd. Accessed 09 June 2019
18. Cao, Z., Hidalgo, G., Simon, T., Wei, S., Sheikh, Y.: OpenPose: realtime multi-person 2D pose estimation using part affinity fields. arXiv preprint arXiv:1812.08008 (2018)
19. Bradski, G.: The OpenCV library. Dr. Dobb's J. Softw. Tools **25**, 120–125 (2000)
20. Pele, O., Werman, M.: Fast and robust earth mover's distance. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 460–467. IEEE (2009)
21. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.A.M.T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
22. Eiben, A.E., Selmar, K.S.: Parameter tuning for configuring and analyzing evolutionary algorithms. Swarm Evol. Comput. **1**(1), 19–31 (2011)