Crowd-driven Mid-scale Layout Design

Tian Feng[1]* Lap-Fai Yu[2] Sai-Kit Yeung[1] KangKang Yin[3] Kun Zhou[4]
[1]Singapore University of Technology and Design [2]University of Massachusetts Boston
[3]National University of Singapore [4]Zhejiang University

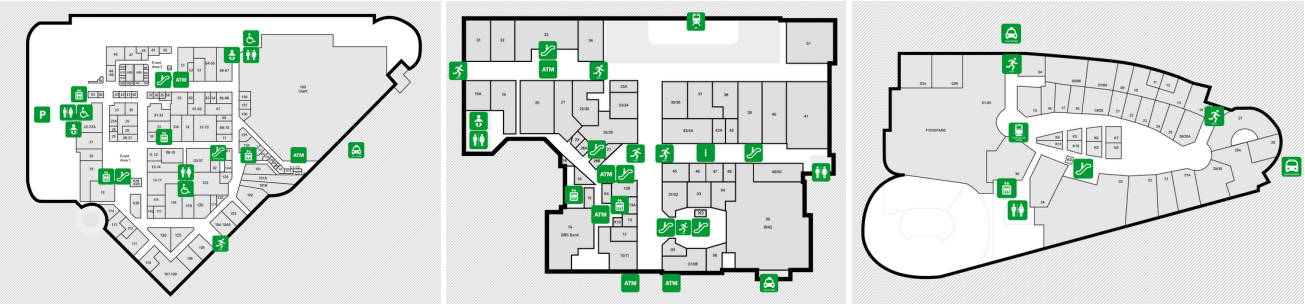


Figure 1: *Real world mall examples.*

Abstract

In this supplementary material, we also include for reference several real world shopping mall examples and pictures of example mall data used for training. We provide further details of the simulation model, the training of random forest regressors and the optimization process. We also provide time usage statistics and further justify the use of regression for approximating full crowd simulation. Finally, we provide a screenshot of our user interface, and further details about the user syntheses described in our main paper, and the example of using our tool for remodeling a real world shopping mall.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric Algorithms

1 Real-world Mall Examples

Figure 1 shows some real world mall examples for reference. Each mall consists of an input layout domain, sites of different types (including facilities) and paths connecting the sites. We define our problem domain similarly. Figure 2 shows some real world mall examples after being digitized by ArcGIS into standard spatial data format (Shapefiles), which we use as our training data.

2 Simulation Model

Agent Model and State Machine. We define our agent models following similar agent models that have been used in the literature [Tu and Terzopoulos 1994; Shao and Terzopoulos 2005; Narain et al. 2009; Li et al. 2012], with slight modifications to suit our purpose. In our case, the agents are used for getting the metrics from the environment necessary for evaluating the agent-based costs, rather than for generating a sophisticated artificial life simulation that aims at modeling the full complexity of human perception (please refer to the work of Shao and Terzopoulos [2005] for a successful attempt along that direction). Note that increasing the complexity of the perception model will increase the computational complexity of the agent-based simulation and hence the time needed for generating training data. As our illustrative example focuses on a shopping mall, we use a state machine inspired by and similarly applied in previous studies in shopping behaviors [Ali and Moulin 2005; Ali 2006; Castillo et al. 2009]. The above usage aims to give a straightforward, first illustration, and does not presume users to have prerequisite knowledge of advanced crowd simulation models. We note that the agent model and state machine can be replaced by users to tackle other specific scenarios if necessary, similar as in common agent-based crowd simulation software such as AnyLogic. In our main paper, we also demonstrate that our crowd simulation model can be replaced by two other popular crowd simulation models, followed by quantitative evaluations which show improvement.

Agent Categories. Marketing research [Bloch et al. 1994] finds that shoppers generally fall into four categories depending on their shopping habits:

- Enthusiasts: shoppers who engage in a wide range of activities (purchasing, experiential consumption, using the mall's facilities).
- Traditionalists: shoppers who only purchase, but not dine or use other services of the mall.
- Minimalists: shoppers who have low average participation in all kinds of activities in the mall.
- Grazers: shoppers who spend more time on eating and taking a rest. They also tend to purchase more than the average level.

Our agent model and state machine can model each of the four agent categories. Please refer to the main paper and video for the effects of changing the proportion of agents belonging to particular categories on the optimized layout designs.

Coziness. Closely following studies in consumers' psychology [Machleit et al. 1994; Ng 2003], we define our coziness cost C_c to penalize either an extremely low or high crowd density, as both degrade the perceived level of coziness. This definition allows our simulation model to mimic realistic situations as reported in literature [Machleit et al. 1994; Ng 2003]. For example, during peak hours, agents visiting a very crowded boutique will feel uncozy. On the other hand, when a mall is closing (i.e., near the end of a simulation), the few remaining agents visiting a nearly-empty boutique will also feel uncozy.

Entering and Leaving Sites. Our simulation model ignores inter-agent collisions as we focus on the crowd flow effects with respect to the layout design rather than the individual agent motions. Our model also ignores the low-level detail of a site's door size with respect to an agent's size, as our work focuses on the optimization

* Part of the work was done when Tian was visiting UMass Boston.

Figure 2: *Example mall data used for training.*

of paths and sites. In our simulation, agents can enter or leave a site simultaneously in case their entering or leaving times coincide.

3 Training and Optimization

Random Forest Regressors. We trained our random forest regressors for predicting the agent-based costs using layout features and full simulation costs obtained from real world layouts. We used the implementation of random forest regressors provided by scikit-learn in Python. We empirically set the number of trees as 10, which gives the best accuracy in our cross validation experiments by repeated random sub-sampling. Please refer to the literature [Liaw and Wiener 2002] for further details of the random forest regression technique.

Simulated Annealing. We optimize our layouts using the simulated annealing technique with a Metropolis-Hastings state-searching step. Given an input layout domain, the layout ϕ is initialized with a few user-specified or randomly-placed entrances on the domain boundary, while a random parameter controls the existence of an atrium. Paths are then generated to connect the entrances and the atrium (if it exists), while splitting the layout domain into several separate sites.

Time Interval	Mobility	Accessibility	Coziness	Time Usage
1 s	0.7401	0.4129	0.8812	470.77 s
5 s	0.7873	0.3010	0.8068	105.31 s
10 s	0.6631	0.2766	0.6107	45.90 s
20 s	0.5232	0.2451	0.6029	18.93 s
30 s	0.5698	0.2449	0.5352	11.08 s

Table 1: *Costs obtained and simulation time usage of running a crowd simulation on the original layout in Figure 14(a), using different time intervals. The costs obtained from using a time interval of 1 second serves as the baseline.*

	Training	Synthesis 1	Synthesis 2	Synthesis 3
Mall 1	20 s	297.23 s	312.89 s	303.42 s
Mall 2	17 s	172.57 s	185.34 s	169.96 s
Theme Park	20 s	218.57 s	198.09 s	192.36 s
Train Station	19 s	236.98 s	239.62 s	202.21 s
Campus	20 s	189.32 s	178.48 s	186.02 s

Table 2: *Training and synthesis time usage for different layouts in Figure 12 in the main paper.*

Our optimization proceeds iteratively. At each iteration, a randomly selected type of move is proposed to generate a proposed layout ϕ' from the current layout ϕ. The total cost function $C(\phi')$ of the proposed layout ϕ' is compared with the total cost function $C(\phi)$ of the current layout ϕ. If the proposed layout has a lower cost ($C(\phi') < C(\phi)$), the current layout is updated to become the proposed layout. Otherwise, the current layout is updated to become the proposed layout only at an acceptance probability. The acceptance probability is high at the beginning of the optimization (corresponding to a high temperature in simulated annealing), allowing the optimizer to quickly explore the solution space; the acceptance probability is low at the end of the optimization (corresponding to a low temperature in simulated annealing), allowing the optimizer to refine the solution (i.e., the layout). We set the temperature of our simulated annealing to gradually decrease over iterations. For further details of the Metropolis Hastings and simulated annealing techniques, please refer to the classic literature [Metropolis et al. 1953; Hastings 1970; Kirkpatrick et al. 1983].

Probabilities of Moves. For simplicity, we use equal probabilities for all types of moves. In our illustrative example (Figure 9 in the main paper), the initial big sites are generally split into smaller sites because of two reasons. First, the coziness cost causes big sites with less visitors to become smaller. Second, a prior total number of sites is set in a prior cost, and as the total area is fixed, the larger the prior total number of sites is, the smaller the sites will become.

4 Time Usage

Time Intervals. By default, we use a time interval of 1 second in our simulation. We avoid using a long time interval as this will result in inaccurate navigation and perception data being captured by the agents.

For example, using a long time interval of 30 seconds, we would frequently miss the important moments when an agent turns around a corner or walks through an intersection. For instance, at one time frame, an agent may be walking towards an crowded intersection. At the next time frame (which refers to 30 seconds later), the agent may have already passed through the intersection. In other words, in both time frames, the agent would not have experienced any crowding which it should have experienced had we used a short time interval. We would *wrongly conclude* from this agent's navigation experience that the layout is good.

Using a short time interval requires a longer simulation time. We decide to trade off time for better accuracy, because the simulation only needs to be done once offline, to create reliable data for training the regressors.

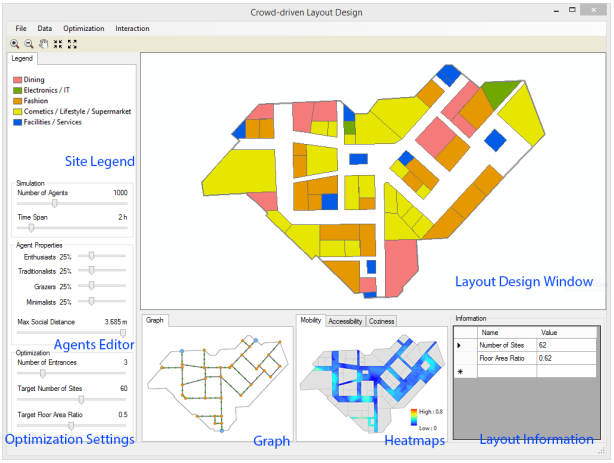


Figure 3: *Our user interface.*

As a test, we conducted simulations using different time intervals, in the original mall layout in Figure 14(a) of the main paper. Table 1 shows the time needed and the costs returned by each simulation. The costs returned by a time interval of 1 second are the most reliable and serve as the baseline. We can see that using a long time interval (e.g. 30 second) generally results in lower costs. The reductions in the costs are due to the fact that the negative effects of crowding are not captured well under a long time interval.

Training and Synthesis Time. Table 2 lists the training and synthesis time for different layouts shown in Figure 12 of the main paper. The training time (excluding the simulation time) for all layouts is about the same (around 20 seconds) using the random forest regressors implementation in Python.

Justification for Approximation. Even using a popular agent-based crowd simulation technique [Narain et al. 2009] optimized for speed, it is difficult to directly drive optimization by simulations and achieve interactive performance. A typical simulation involves at least 1000 agents (we also used 2000 and 3000 agents for bigger layouts as mentioned in our experiments) navigating in a layout for 2 simulated hours with regular time interval being 1 second, corresponding to about 7.2 million agent updates in total. This simulation would take about 0.5 minute for the aforementioned fast technique to finish, which is able to make about 0.2 million agent updates per second according to the reported statistics of the technique. An optimization requires about 300 such simulations, and hence would require about 150 minutes (2.5 hours) to finish even using this popular crowd simulation technique optimized for speed.

Note that the above statistics are an estimate. Our crowd simulation model is more computationally expensive because we also need to update each agent's perception, internal states and navigation goals in addition to its position. The simulation time needed also depends on the complexity of the layout. For our running example (*Mall 1*), using 1000 agents, 2 simulated hours and 1 second time interval, one full simulation by our crowd simulation model takes about 3 minutes. One layout optimization typically requires about 300 iterations, and hence takes about 15 hours to finish if a full simulation is run at each iteration.

5 Interface

We implemented our main approach as an interactive layout modeling tool, using C# and the .NET Framework. Figure 3 shows a screenshot of our interface.

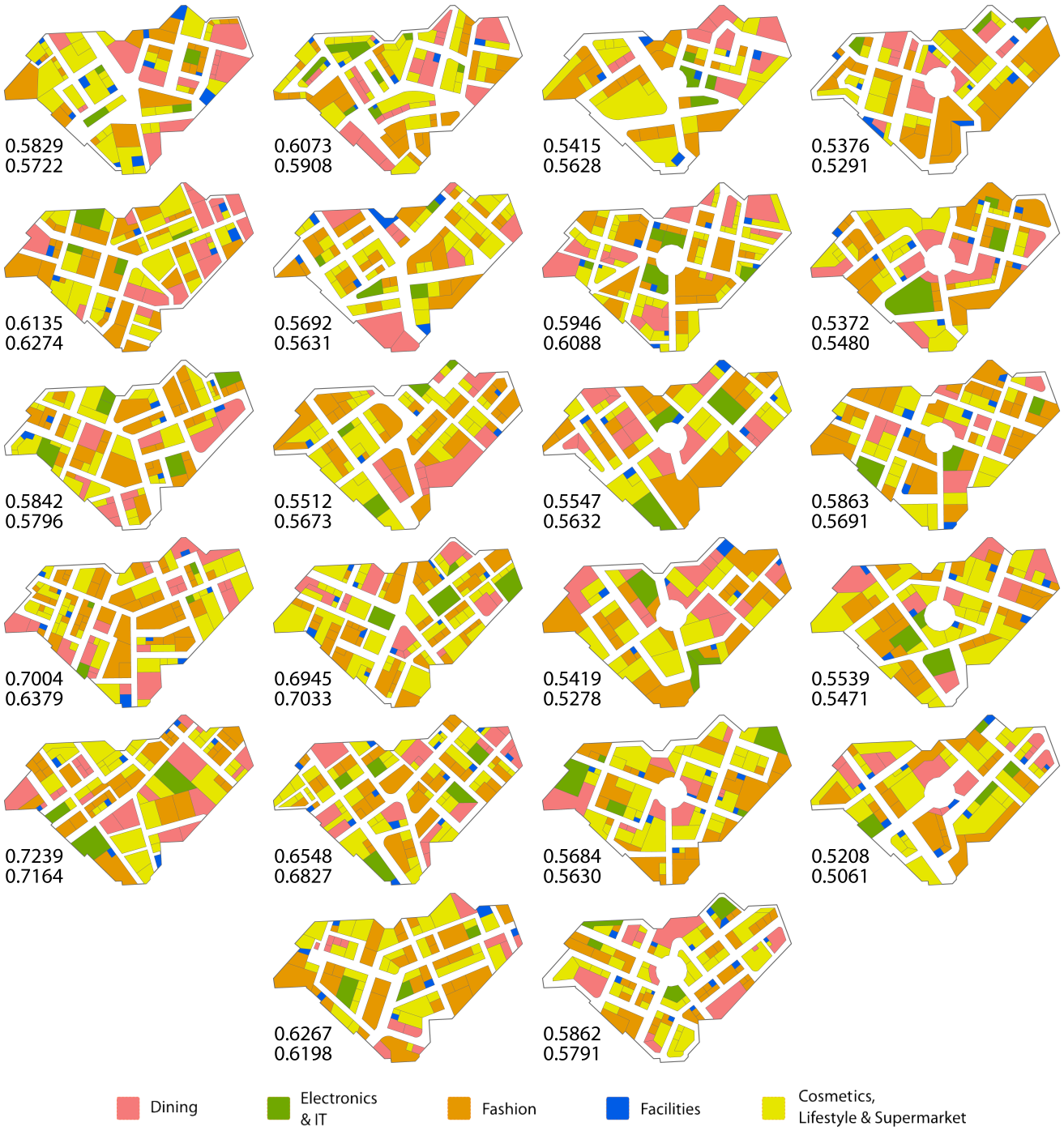


Figure 4: *User-synthesized layouts of Mall 1. For each synthesis, the numbers at the top and at the bottom are the average crowd densities from AnyLogic and PathFinder respectively. The real world layout of Mall 1 has average crowd densities of* 0.9348 *and* 0.9069 *from AnyLogic and PathFinder respectively.*

6 User Synthesis

We invited 22 senior undergraduate students in architectural design to use our tool for synthesizing layouts given the boundary of *Mall 1* and *Mall 2* mentioned in the main paper. Figure 4 and Figure 5 show the synthesized layouts from users.

7 Remodeling

We use our approach to remodel a real world shopping mall (Figure 6(a)), which is badly reputed for its inconvenience in internet discussion [2007]. The original mall shows a poor design with narrow paths, cramped stores and an inconvenient distribution of shops and facilities, which gives a high cost in our full simulation. Our approach remodels this mall automatically. The remodeled mall shows less congestion and is more cozy to visit. Figure 6(b)–(d) visualize the substantial improvement in all the agent-based costs. Figure 7 shows the evaluation on the original and the remodelled layouts by AnyLogic and PathFinder. Refer to the figure caption for details.

Figure 5: *User-synthesized layouts of Mall 2. For each synthesis, the numbers at the top and at the bottom are the average crowd densities from AnyLogic and PathFinder respectively. The real world layout of Mall 2 has average crowd densities of* 0.8391 *and* 0.9130 *from AnyLogic and PathFinder respectively.*

8 Prior Costs

In Equation (1) of our paper, $\mathbf{C}_{\mathrm{P}} = [C_{\mathrm{pr}}^{\mathrm{total}}, C_{\mathrm{pr}}^{\mathrm{type}}, C_{\mathrm{pr}}^{\mathrm{ratio}}]$ stores the prior costs which encode the priors specific to the type of the layout to be synthesized, and $\mathbf{w}_{\mathrm{P}} = [w_{\mathrm{pr}}^{\mathrm{total}}, w_{\mathrm{pr}}^{\mathrm{type}}, w_{\mathrm{pr}}^{\mathrm{ratio}}]$ stores their weights. The prior costs are defined as follows.

Prior Total Number of Sites. This cost penalizes if the total number of sites in the current layout ϕ deviates from the prior total number of sites:

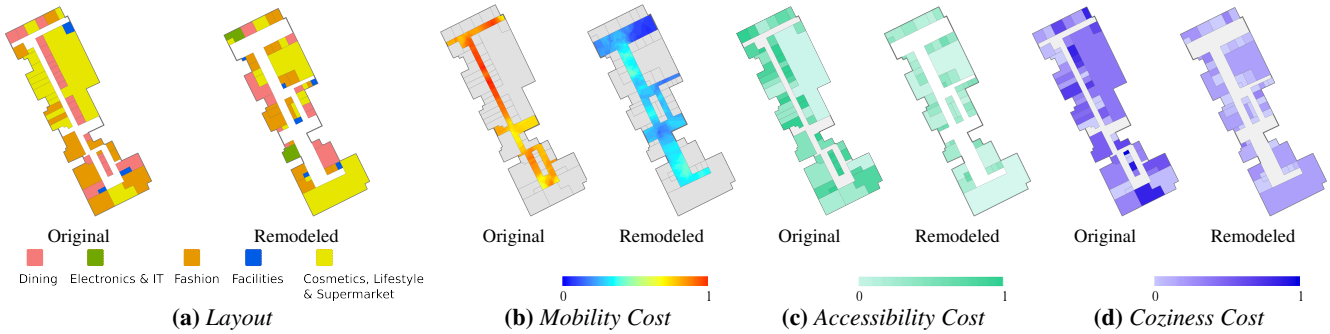
$$C_{\mathrm{pr}}^{\mathrm{total}}(\phi) = \frac{1}{Z_{\mathrm{pr}}^{\mathrm{total}}} \left| s_{\mathrm{total}} - s'_{\mathrm{total}} \right|, \tag{1}$$

where s_{total} is the total number of sites in layout ϕ; s'_{total} is the prior total number of sites; $Z_{\mathrm{pr}}^{\mathrm{total}}$ is a normalization constant which we set to 100.

Prior Number of Each Type of Sites. This cost penalizes if the number of sites of a particular type deviates from the prior number of that type of sites:

$$C_{\mathrm{pr}}^{\mathrm{type}}(\phi) = \frac{1}{Z_{\mathrm{pr}}^{\mathrm{type}}} \sum_i \left| s_i - s'_i \right|, \tag{2}$$

where s_i is the number of sites of type i in layout ϕ; s'_i is the prior



(a) *Layout* **(b)** *Mobility Cost* **(c)** *Accessibility Cost* **(d)** *Coziness Cost*

Figure 6: *Remodeling of a real world shopping mall. The remodeled mall shows improvement in all the agent-based costs. (a) Layouts of the original and the remodeled mall. (b) The mobility cost helps make the paths wider and also move the circular paths to the middle to facilitate the flow of human movement. An agent in the middle can easily turn around to visit a site at either side of the mall. (c) The accessibility cost ensures an even distribution of facilities such as restrooms. (d) The coziness cost prompts the huge Cosmetics, Lifestyle & Supermarket site in the original mall to split into two medium-sized sites, distributed at both sides of the mall, such that shoppers can access either one easily.*

number of sites of type i; $Z_{\text{pr}}^{\text{type}}$ is a normalization constant which we set to 100.

Prior Ratio of Sites' Area to Layout Area. This cost penalizes if the ratio of sites' area to layout area (commonly known as floor-area ratio) deviates from the prior ratio:

$$C_{\text{pr}}^{\text{ratio}}(\phi) = \frac{1}{Z_{\text{pr}}^{\text{ratio}}} \left| r - r' \right|, \tag{3}$$

where r is the ratio of sites' area to layout area in layout ϕ; r' is the prior ratio; $Z_{\text{pr}}^{\text{ratio}}$ is a normalization constant which we set to 1.

Cost Weights. Unless otherwise specified, we set our cost weights as $w_{\text{m}} = 1.0, w_{\text{a}} = 1.0, w_{\text{c}} = 1.0, w_{\text{pr}}^{\text{total}} = 0.5, w_{\text{pr}}^{\text{type}} = 0.5$, and $w_{\text{pr}}^{\text{ratio}} = 0.5$.

Each of our costs is normalized to the range $[0.0, 1.0]$. Under the current weighting scheme, the total cost function $C(\phi)$ and the approximated total cost function $\hat{C}(\phi)$ stay in the range $[0.0, 4.5]$.

9 Layout Features

To use random forest regressors [Liaw and Wiener 2002] for approximating the agent-based costs, we extract from a layout a number of geometrical, topological and general features defined as follows.

9.1 Geometric Features

Edge Length. Maximum, minimum, mean and standard deviation of the lengths of all edges in the graph representing the layout.

Edge Width. Maximum, minimum, mean and standard deviation of the widths of all edges in the graph representing the layout.

Site Area. Maximum, minimum, mean and standard deviation of the areas of all sites.

9.2 Topological Features

Node Valence. Mean and normalized histogram of the valences of all nodes in the graph representing the layout. The valence of a node is defined as the number of its connected edges.

Edge Valence. Maximum, minimum, mean and standard deviation of the valences of all edges in the graph representing the layout. For an edge e with start node v_i and end node v_j, its valence is defined as:

$$val(e) = val(v_i) + val(v_j) - 2, \tag{4}$$

where $val(v_i)$ and $val(v_j)$ are node valences for v_i and v_j.

Travel Distance. Maximum, minimum, mean and standard deviation of the distances of the routes from ten randomly-selected nodes to their nearest site of each type.

Travel Convenience. Maximum, minimum, mean and standard deviation of the travel convenience of the routes from ten randomly-selected nodes to their nearest site of each type. For a route ω with start node v_i and end node v_j, its travel convenience is defined as:

$$tc(\omega) = \frac{\lambda_{\text{e}}(\omega)}{\lambda(\omega)}, \tag{5}$$

where $\lambda_{\text{e}}(\omega)$ is the Euclidean distance between v_i and v_j; $\lambda(\omega)$ is the total length of route ω.

Betweenness Centrality. Maximum, minimum, mean and standard deviation of the betweenness centralities of all nodes in the graph. To compute the betweenness centralities, first we calculate the shortest paths between every pair of nodes. For each node, its betweenness centrality is defined as the number of times it appears on a shortest path.

Closeness Centrality. Maximum, minimum, mean and standard deviation of the closeness centralities of all nodes in the graph. To compute the closeness centrality of a node, first we calculate the shortest path between the node and all other nodes. Its closeness centrality is defined as the sum of the reciprocals of the shortest path distances.

9.3 General Features

Site Histogram. Normalized histogram of the number of sites of all types.

Path Area Ratio. The sum of areas covered by all paths relative to the area of the layout domain:

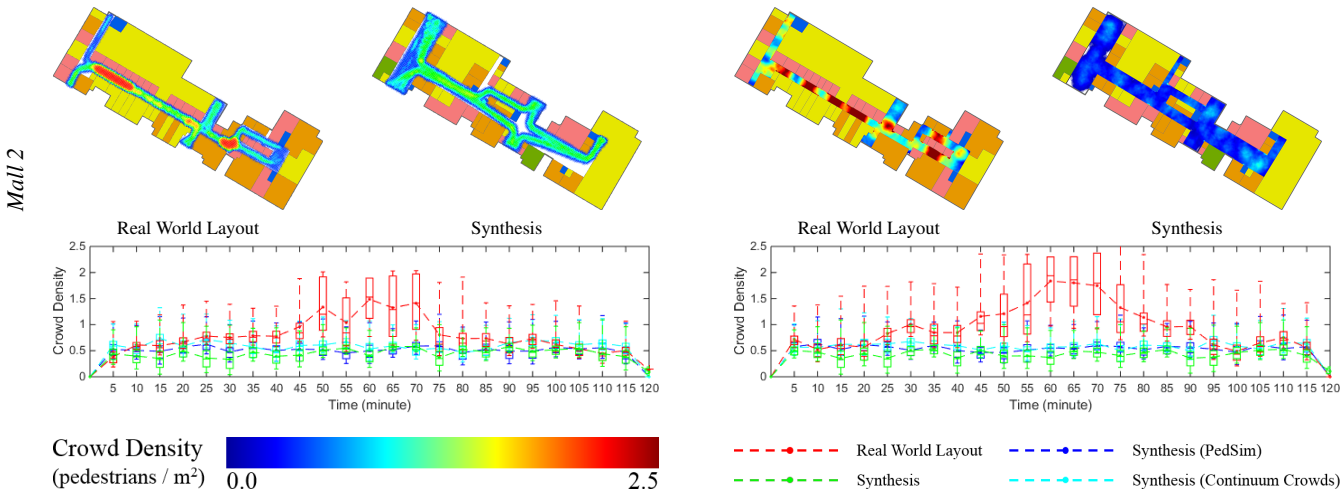
$$par = \frac{1}{\alpha} \sum_{p} Area(p), \tag{6}$$

where α is the area of the layout domain; $Area(p)$ is the area covered by path p.

Agent Density. The average number of agents over time:

$$ad = \frac{N}{T}, \tag{7}$$

where N is the total number of agents in the layout and T is the length of time that the crowd simulation spans.



(a) *AnyLogic Results* **(b)** *PathFinder Results*

Figure 7: *Evaluating real world layout and our syntheses of Mall 2 by (a) AnyLogic and (b) PathFinder. Heat maps show crowd densities at the middle of the simulation. Curves and box plots show the means and statistics of crowd densities versus simulation time.*

References

ALI, W., AND MOULIN, B. 2005. 2d-3d multiagent geosimulation with knowledge-based agents of customers' shopping behavior in a shopping mall. In *Spatial Information Theory*, A. Cohn and D. Mark, Eds., vol. 3693 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 445–458.

ALI, W. 2006. *Developping 2D and 3D multiagent geosimulation, a method and its application: the case of shopping behavior geosimulation in Square One Mall (Toronto)*. PhD thesis, Laval University.

BLOCH, P. H., RIDGWAY, N. M., AND DAWSON, S. A. 1994. The shopping mall as consumer habitat. *Journal of Retailing 70*, 1, 23 – 42.

CASTILLO, L. F., BEDIA, M. G., URIBE, A. L., AND ISAZA, G. 2009. A formal approach to test commercial strategies: Comparative study using multiagent based techniques. *Journal of Physical Agents 3*, 3, 25–30.

HASTINGS, W. K. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika 57*, 1 (Apr.), 97–109.

KIRKPATRICK, S., GELATT, C. D., AND VECCHI, M. P. 1983. Optimization by simulated annealing. *SCIENCE 220*, 4598, 671–680.

LI, W., DI, Z., AND ALLBECK, J. 2012. Crowd distribution and location preference. *Computer Animation and Virtual Worlds 23*, 3, 343–351.

LIAW, A., AND WIENER, M. 2002. Classification and regression by randomforest. *R News 2*, 3, 18–22.

MACHLEIT, K. A., KELLARIS, J. J., AND EROGLU, S. A. 1994. Human versus spatial dimensions of crowding perceptions in retail environments: A note on their measurement and effect on shopper satisfaction. *Marketing Letters 5*, 2, 183–194.

METROPOLIS, N., ROSENBLUTH, A. W., ROSENBLUTH, M. N., TELLER, A. H., AND TELLER, E. 1953. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics 21*, 6, 1087–1092.

NARAIN, R., GOLAS, A., CURTIS, S., AND LIN, M. C. 2009. Aggregate dynamics for dense crowd simulation. *ACM Trans. Graph. 28*, 5 (Dec.), 122:1–122:8.

NG, C. F. 2003. Satisfying shoppers' psychological needs: From public market to cyber-mall. *Journal of Environmental Psychology 23*, 4, 439–455.

SGFORUMS, 2007. What's the best & worst shopping center in singapore? http://sgforums.com/forums/8/topics/230251. Accessed: 01-01-2016.

SHAO, W., AND TERZOPOULOS, D. 2005. Autonomous pedestrians. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, New York, NY, USA, SCA '05, 19–28.

TU, X., AND TERZOPOULOS, D. 1994. Artificial fishes: Physics, locomotion, perception, behavior. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, SIGGRAPH '94, 43–50.