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

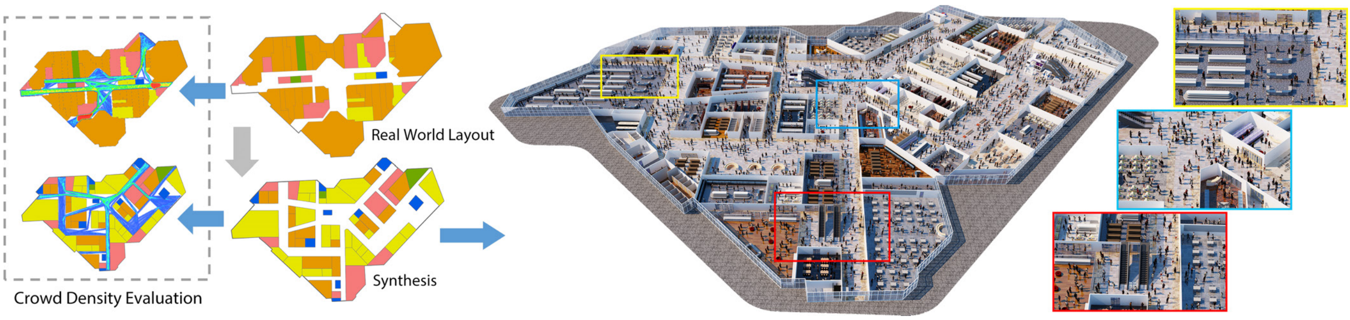


Figure 1: *Given a layout domain from a real world layout, our approach synthesizes crowd-aware layouts by considering the crowd flow properties of visitors: mobility, accessibility and coziness. Evaluation by crowd simulation software shows that our synthesized layouts exhibit improved crowd flow properties compared to the input real world layouts. Three-dimensional visualization on the right shows the flow of human crowds in a shopping mall created using the synthesized layout on the left.*

Abstract

We propose a novel approach for designing mid-scale layouts by optimizing with respect to human crowd properties. Given an input layout domain such as the boundary of a shopping mall, our approach synthesizes the paths and sites by optimizing three metrics that measure crowd flow properties: mobility, accessibility, and coziness. While these metrics are straightforward to evaluate by a full agent-based crowd simulation, optimizing a layout usually requires hundreds of evaluations, which would require a long time to compute even using the latest crowd simulation techniques. To overcome this challenge, we propose a novel data-driven approach where nonlinear regressors are trained to capture the relationship between the agent-based metrics, and the geometrical and topological features of a layout. We demonstrate that by using the trained regressors, our approach can synthesize crowd-aware layouts and improve existing layouts with better crowd flow properties.

Keywords: layout design, agent-based crowd simulation

Concepts: •**Computing methodologies** → **Modeling and simulation;** •**Theory of computation** → *Computational geometry;*

> *"The aesthetic of architecture has to be rooted in a broader idea about human activities like walking, relaxing and communicating. Architecture thinks about how these activities can be given added value."*
>
> — Thom Mayne

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

SIGGRAPH '16 Technical Paper, July 24-28, 2016, Anaheim, CA,
ISBN: 978-1-4503-4279-7/16/07
DOI: http://dx.doi.org/10.1145/2897824.2925894

1 Introduction

Computational layout design methods have gained more and more research attention in recent years [Parish and Müller 2001; Michalek et al. 2002; Aliaga et al. 2008; Vanegas et al. 2009; Merrell et al. 2010; Peng et al. 2014; Peng et al. 2015]. The main advantage of such methods is that they enable users to focus on high-level design specifications while shifting the tedious low-level operations to computers. Among various factors governing a good layout design, the human factor is critically important, because human activities and architectural layout designs are closely coupled. As Pritzker Prize-winning architect Thom Mayne further noted, "I absolutely believe that architecture is a social activity that has to do with some sort of communication or places of interaction, and that to change the environment is to change behavior." Examples supporting this observation are evident in everyday's layout designs. A well-designed, highly-accessible theme park, where attractions, facilities and pathways are conveniently laid out, can lead to high satisfaction of visitors of different age groups [Niles 2012; Clave 2007]. In contrast, a badly-designed shopping mall, where shops and facilities are faultily-connected, can result in frustrating visiting experience and low usage [D. Beyard et al. 2006; Brown 1999].

Previous methods for layout design have mainly focused on either large-scale urban planning (e.g., city layout design) where human factors are considered in a macroscopic level such as distribution of population and jobs [Vanegas et al. 2009], or small-scale residential design (e.g., furniture layout design) where the focus is on generating a decent living environment (such as a living room) for individuals [Yu et al. 2011; Merrell et al. 2011]. In neither case are human crowd properties the major considerations. The design of mid-scale environments and social places such as shopping malls, theme parks, train station buildings and university campuses, where large crowds of visitors meet and move about, are relatively less studied. We dub this type of layouts with a scale intermediate between the scales of urban and residential layouts, and where human activities represented by crowd flows are the major considerations, *mid-scale layouts*. The goal of this paper is to put human activities back into the layout design loop, and synthesize *crowd-aware* mid-scale layouts that are optimal with respect to the flow of human crowds.

We propose to focus on aggregate human crowd flow properties, such as mobility and comfort, for layout designs of mid-scale environments by using crowd simulation. Crowd simulation is a standard tool in evaluating the safety level offered by a manual design in engineering. In this work, we take a step further by using the metrics obtained from crowd simulation to design and optimize a layout automatically. Note that even in the state-of-the-art crowd simulation software (e.g., PathFinder [2016], AnyLogic [2016], MASSIVE [2016]), while the simulation results could inform the planner about the potential problems inherent in the layout design, the modifications necessary for improving the design still largely rely on the planner's manual efforts and experience. Such modifications can be difficult to make in view of the complex human-layout interaction. We propose an optimization framework that takes into account the crowd flow properties, namely, mobility, accessibility and coziness, extracted from an agent-based crowd simulation.

Each full agent-based crowd simulation, however, can take up to minutes to compute. It would be computationally too costly to evaluate the agent-based metrics directly from a full simulation for every iteration of the optimization. To make our approach more practical, we propose a novel data-driven regression method to capture the relationship between the agent-based costs, and the geometric and topological features of the layout. This way, we can instead optimize against the approximated agent-based costs predicted by the trained regressors, and thus achieve interactive performance. Our experiments show that approximating by regressors leads to a speedup of two orders of magnitude, while achieving similar performance and end results as using full simulation, hence making it a practical choice for incorporating human crowd considerations into a layout design tool.

Layout synthesis and crowd simulation are both important areas in computer graphics. By proposing *crowd-driven layout design*, we hope to integrate the merits from both areas in order to generate layouts not only optimal with respect to architectural design principles or functional requirements, but also comfortable for human beings or digital avatars who will eventually visit these facilities in real life or in virtual worlds.

The major contributions of our work are:

- introducing a novel approach for generating crowd-aware mid-scale layouts optimized with respect to human crowd flow properties: mobility, accessibility, and coziness. These properties are captured by novel agent-based costs evaluated through crowd simulations.
- circumventing the expensive computations of agent-based costs by approximating the relationships between human crowd flow properties and various layout features with the outputs of pre-trained non-linear regressors. The approximation enables our approach to synthesize layouts quickly and to support interactive layout design.
- demonstrating the novel applications of the proposed approach for automatically generating a variety of crowd-driven layout designs, changing layout styles with respect to agent properties, and remodeling and improving existing layouts.

2 Related Work

Recent work in layout design has focused on automatic space planning and allocation problems, which are often approached in two different scales in graphics: urban-scale and residential-scale.

Urban-scale Layout Modeling. Urban-scale modeling spans a wide range of topics across computer graphics and vision, including geometry acquisition, layout modeling, street modeling, facade modeling, simulation and behavioral modeling, etc. Please refer to the survey paper [Aliaga 2012] for a good overview. In this paper, we focus on the literature about layout modeling, i.e., the spacial arrangement of layouts.

To design an urban layout, most existing computational approaches follow a *streets-then-parcels* paradigm: first generate a street network; then subdivide the blocks surrounded by streets into parcels. Parish and Müller [2001] first adopted this approach to generate urban streets and parcels using L-systems [Prusinkiewicz and Lindenmayer 1996]. Sun et al. [2002] proposed a rule-based rewriting approach to generate road networks for city modeling. Galin et al. [Galin et al. 2010; Galin et al. 2011] applied procedural modeling to generate hierarchical road networks. Weber et al. [2009] proposed a street expansion algorithm to simulate three-dimensional urban models over time. Aliaga et al. [2008] presented an urban layout synthesis algorithm which utilizes attributed intersection points and aerial-view images. Chen et al. [2008] used tensor fields to design and edit urban models and guide street generation. Nishida et al. [2015] synthesized road networks from examples. Beneš et al. [2014] grew procedural road networks for cities by simulating traffic with neighboring cities. Vanegas et al. [2009] devised an interactive system using geometrical and behavioral modelling for designing urban spaces, and later on a procedural modeling system for generating geometrically and functionally plausible parcels [Vanegas et al. 2012b].

More recently, Yang et al. [2013] added global optimization to their hierarchical splitting method to guarantee the fairness and regularity of the generated streets and parcels. Different from the above methods, Peng et al. [2014] tiled a layout domain with deformable meshes which are geometrically optimized to further reduce shape registration errors, without producing streets in advance. Their subsequent work [Peng et al. 2015] focuses on computationally creating layouts starting from high-level functional specifications, by posing an integer programming-based approach driven by topological constraints and scores.

Compared to the previous works, a major novelty of our approach is layout optimization with respect to different human criteria, for example, ease of movement and comfort. While it is difficult to encode these human criteria in procedural rules, evaluating them using an agent model is straightforward. Our approach allows a user to synthesize people-oriented layout designs, by intuitively specifying agent properties, without the need to manipulate procedural rules.

Most related to our work is the urban procedural modeling system proposed by Vanegas et al. [2012a]. However, their objective is to generate a urban-scale city layout populated with thousands of buildings, while our objective is to generate a mid-scale layout comprising of functional sites (e.g., shops, facilities) where humans can navigate and socialize. Their system are optimized against macroscopic criteria such as sunlight exposure and landmark visibility which are important for urban-scale (outdoor) layout design. In contrast, we incorporate human navigation and perception with agent-based models, and optimize against crowd properties such as mobility and coziness which are important for mid-scale (mainly indoor) layout design. In addition, they trained their neural network to adjust parameters of procedural rules to achieve desirable layouts corresponding to given indicators, while the parameters of our optimization and simulation stay fixed. We train regressors to approximate agent-based costs originally obtained by full crowd simulation in order to generate crowd-aware layouts at interactive rates.

Residential-scale Layout Modeling. Traditional approaches focused on exhaustively searching for possible layout designs over a certain number of rooms [Galle 1981], or over all partitions of a grid [Shaviv 1974]. Michalek et al. [2002] proposed an evolutionary algorithm to generate layouts for single-story, rectangular apartments, using connectivity relationships between rooms. Harada et al. [1995] presented an approach which supports interactive manipulation of architectural layouts by shape grammars.

More recently, Merrell et al. [2010] devised an approach to generate residential floor-plans from a set of high-level user requirements. Liu et al. [2013] proposed a pipeline for generating floor plans of precast concrete-based residences by conforming to functional, design and fabrication constraints. Yu et al. [2011] proposed an automatic approach and Merrell et al. [2011] devised an interactive tool for designing furniture layouts. Fisher et al. [2015] synthe-

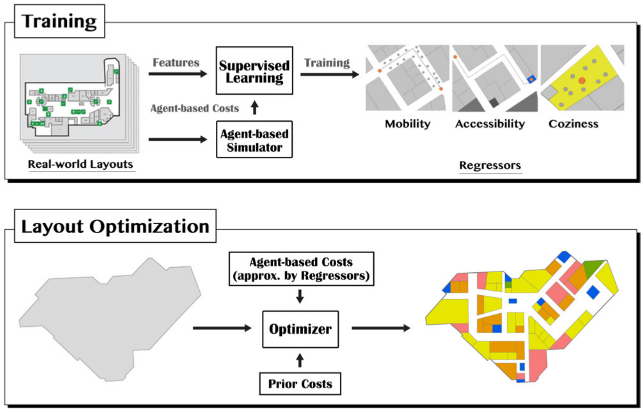
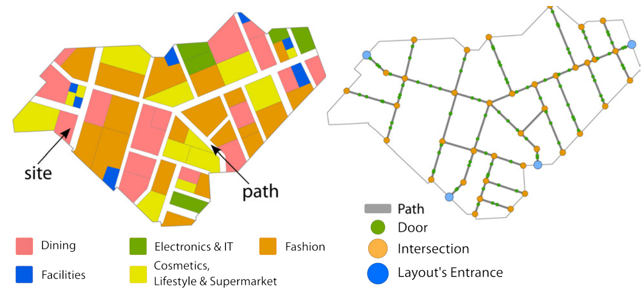


Figure 2: *Overview of our approach.*



(a) *Layout* **(b)** *Graph*

Figure 3: *The layout of a shopping mall used as our illustrative example.*

sized functional 3D scenes by deducing possible human activities. Most recently, AlHalawani and Mitra [2015] proposed an approach for optimizing object placement in a warehouse by analyzing traffic congestion. Differently from these approaches, our work focuses on using human flow properties such as coziness and mobility for synthesizing mid-scale architectural layouts.

Crowd Simulation for Layout Evaluation. Crowd simulation is an important topic in computer graphics and virtual reality, mainly for generating realistic animations of crowds or populating virtual environments with realistic agents. We refer the interested readers to a few excellent surveys for a comprehensive overview [Pettré et al. 2008; Huerre et al. 2010]. Crowd simulation has also been used extensively for the analysis, evaluation, and visualization of building designs and urban plans in engineering and architecture [Li and Liu 2008; Huerre et al. 2010; Aschwanden et al. 2011]. The crowd simulation results help architects and urban planners make informed decisions about their design with humans in the loop, as the ultimate criteria of a good design is to simultaneously improve the quality of life and the sustainability of the created environment. Among the various simulation methods available in the literature, agent-based simulation methods are commonly used for layout evaluation due to their simplicity, generality and flexibility. We first demonstrate our framework with an agent-based simulation method. We then show that our approach can employ two more advanced simulation methods as well, namely, *PEDSIM* [2016], which is based on the social force model [Helbing and Molnar 1995], and *Continuum Crowds* [2006], which is based on continuum dynamics.

Quite unique to our work is that crowd simulation is utilized not only for evaluating a final layout design, but also for generating the design automatically and improving the design interactively. This is achieved by integrating the crowd simulation results into an optimization objective. Therefore the optimization framework needs to trigger the simulator numerous times in order to find a good design. For illustrating our framework, we use an agent-based simulation method which models agents' navigation with a state machine. We ignore inter-agent collisions as we mainly care about crowd flow effects rather than individual agent motions. We demonstrate that our framework can be used to synthesize different crowd-aware mid-scale layouts according to different agent properties and visiting habits, which can be easily specified by users.

3 Overview

Figure 2 shows an overview of our approach. Given an input layout domain, e.g., boundary of a shopping mall, our goal is to automatically synthesize crowd-aware mid-scale layouts which are optimal with respect to agents' comfort and ease of movement. This is achieved by optimizing a layout against estimated agent-based costs and user-directed prior costs. The agent-based costs evaluate the experience perceived by agents navigating in the layout, in terms of mobility, accessibility and coziness. The prior costs encode the design goals that influence the layout to be synthesized, such as the floor area ratio. The layout is iteratively updated until it converges to an optimal layout.

Our approach consists of two parts: offline training and online layout optimization. Ideally, the agent-based costs should be computed directly by performing an agent-based simulation in the layout for each iteration of the optimization. However, a typical optimization requires hundreds of iterations and thus takes hours to compute, even with a simplified agent-based model where more sophisticated features such as collision avoidance are turned off. We thus first model the relationship between the geometrical and topological features of layouts and their corresponding agent-based costs by performing nonlinear regression on a database of real world layout examples.

With the trained regressors, we can predict the agent-based costs fast enough with high accuracy. Together with the prior costs which encode different priors such as the desired total number of sites and the desired number of sites of each type, we can optimize a layout in about 2 to 5 minutes instead of hours. In the following, we will first discuss our layout representation. Then we will focus on the agent-based costs and the nonlinear regression. Finally we will detail our stochastic optimization method and the layout modification strategies.

4 Problem Formulation

4.1 Layout Representation

Figure 3 shows a layout of a shopping mall which we use as our illustrative example, and its graph representation. Similar as the floor plans of many common public places (e.g., train station buildings, shopping malls), each layout ϕ comprises a set of sites separated by a set of paths, and a number of entrances where agents can enter or leave the layout.

Site. Each site is represented as a polygon and is associated with a functional type. For example, for a shopping mall, a site can be designated as a restaurant, a shop, or a restroom, etc. Each site also has a few doors accessible from its adjacent paths, through which an agent can enter or leave the site.

Path. Each path is represented as a straight line or a curve. A path can connect to either another path, a door to a site, or a layout's entrance, at its start or end point. A path has a certain length and width, which determine the number of agents it can accommodate.

Graph. To facilitate our subsequent computation, we construct a graph $G = \{V, E\}$ for each layout. Figure 3(b) shows the graph of the example layout. Each node $v \in V$ refers to either an intersection between two paths, a door to a site, or a layout's entrance. Each edge $e \in E$ refers to a path. Two nodes are connected by an edge in the graph if their corresponding elements (e.g., two doors) are connected by a path in the layout.

4.2 Optimization Objective

We formulate our problem as follows. Given an input layout domain (Figure 2), a user-specified boundary within which sites and paths can be generated, we search for a crowd-aware layout ϕ^* by minimizing a total cost function:

$$C(\phi) = \mathbf{C}_{\text{A}} \mathbf{w}_{\text{A}}^T + \mathbf{C}_{\text{P}} \mathbf{w}_{\text{P}}^T, \tag{1}$$

where $\mathbf{C}_{\text{A}} = [C_{\text{m}}, C_{\text{a}}, C_{\text{c}}]$ is a vector of agent-based costs and $\mathbf{w}_{\text{A}} = [w_{\text{m}}, w_{\text{a}}, w_{\text{c}}]$ is a vector of weights. C_{m}, C_{a} and C_{c} encode the agent-based considerations: mobility, accessibility and coziness. $\mathbf{C}_{\text{P}}$ is a vector of prior costs and $\mathbf{w}_{\text{P}}$ stores the weights of these costs. The prior costs encode the priors specific to the type of the layout to be synthesized.

Ideally, the agent-based costs are evaluated based on the agents' perception in an agent-based simulation conducted in the current layout ϕ. However, it is too costly to be practical. Instead, we optimize against the agent-based costs approximated by our regressors, $\hat{\mathbf{C}}_{\text{A}} = [\hat{C}_{\text{m}}, \hat{C}_{\text{a}}, \hat{C}_{\text{c}}]$. The approximated total cost function $\hat{C}(\phi)$ is given by:

$$\hat{C}(\phi) = \hat{\mathbf{C}}_{\text{A}} \mathbf{w}_{\text{A}}^T + \mathbf{C}_{\text{P}} \mathbf{w}_{\text{P}}^T \tag{2}$$

5 Agent-based Simulation

Note that while eventually we will optimize against the regressor-approximated costs $\hat{C}_{\text{m}}, \hat{C}_{\text{a}}$ and $\hat{C}_{\text{c}}$, to train the regressors beforehand, we still need to run crowd simulations on a set of layouts and obtain their agent-based costs C_{m}, C_{a} and C_{c}. We first employ an agent-based simulation model as will be described below. We further test our framework with two other crowd simulation models in Section 9.

5.1 Agent Model

Agents represent human visitors of a layout. As our focus is on using agents to measure the convenience and coziness offered by a layout, rather than to generate a highly realistic crowd simulation for visualization or animation purposes, we use a simple agent model that focuses on the i) navigation behavior (movement), ii) perception of crowding (density) and iii) locomotion (walking speed) of the agent. Similar agent models have been used in the literature [Tu and Terzopoulos 1994; Shao and Terzopoulos 2005; Narain et al. 2009; Li et al. 2012]. But we tailor the model specific to our problem domain of layout design, and we do not consider collision detection nor a sophisticated cognitive model. Table 1 depicts some of our agents' properties. In the following we describe how we model our agents in detail.

5.1.1 Navigation

The navigation behavior of each agent η is controlled by a state machine, akin to the common practice in previous crowd simulation work [Tu and Terzopoulos 1994; Yu and Terzopoulos 2007]. For the illustrative shopping mall example, we use a state machine depicted in Figure 5 to simulate shoppers, inspired by previous studies on shopping behaviors [Ali and Moulin 2005; Castillo et al. 2009]. An agent's state can be either *think*, *walk* or *visit* at any particular time. An agent's state is *think* when it is deciding the next destination site to walk to. Its state is *walk* when it is walking on a path towards a destination site. Once an agent enters a site, its state becomes *visit*. An agent visits a site for a certain length of time which

Navigation	**Example Value**
start time	9:00 am
total length of visit	90 min
visiting sequence	restaurant → shop → restroom
state	*visit*
Perception	
social distance	1.82 m
Locomotion	
age	21
default walking speed	$1.50\,\mathrm{m\,s^{-1}}$
actual walking speed	$0.82\,\mathrm{m\,s^{-1}}$

Table 1: *Properties of an example agent.*

Dining	Electronics & IT	Fashion	Facilities	Cosmetics, Lifestyle & Supermarket
45 min	10 min	15 min	5 min	15 min

Table 2: *Length of visit for different site types.*

depends on the site type. For simplicity, we set a fixed *length of visit* for each site type, as depicted in Table 2.

An agent enters the layout at a certain *start time*, and navigates in the layout for a certain length of time, referred to as the *total length of visit*. Each agent has a *visiting sequence* in mind, which specifies the types of sites the agent plans to visit in a particular order. For example, in a shopping mall layout, an agent may want to go to a restaurant, a shop, a restroom and then leave the layout. The visiting sequence is determined by the category of the agent's visiting habit in a particular environment. For example, marketing research [Bloch et al. 1994] generally classifies shoppers into four categories according to their shopping habits, *enthusiasts*, *traditionalists*, *minimalists* and *grazers* (refer to our supplementary material for details). In synthesizing shopping malls, we thus allow the user to assign agents to one of these four categories. Each category is associated with a different visiting sequence which reflects shopping habits. For example, grazers tend to spend more time on dining.

5.1.2 Perception

In a crowded place, a person can perceive the behavior of others usually only within a certain distance. Most social activities take place within a *social distance* ranging from 1.219 m (4 ft) to 3.685 m (12 ft) between two people [Hall 1990; Narain et al. 2009]. Accordingly, we define the social distance r_η for agent η as:

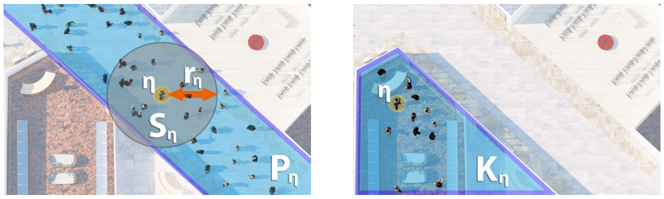
$$r_\eta = r_{\text{min}} + \xi_r (r_{\text{max}} - r_{\text{min}}), \tag{3}$$

where ξ_r is a uniformly distributed random number between 0 and 1; $r_{\text{min}} = 1.219$ and $r_{\text{max}} = 3.658$ are the lower and upper bounds of the social distance. We include the randomness ξ_r as different people may have different social distances and perceptions of crowding [Machleit et al. 2000].

Baseline Crowd Density. If there is no other agent within an agent η's social distance, agent η will perceive only one agent (i.e., itself) within its social distance. In this scenario, agent η perceives a *baseline crowd density* ρ_η^{base} given by:

$$\rho_\eta^{\text{base}} = \frac{1}{r_\eta^2 \pi} \tag{4}$$

Walking Crowd Density. An agent perceives a *walking crowd*



(a) *Walking Crowd Density* **(b)** *Visiting Crowd Density*

Figure 4: *Computing the (a) walking crowd density and (b) visiting crowd density.*

density when it is walking on a path, defined as follows:

$$\rho_\eta^{\text{walk}} = \frac{N_\eta^{\text{walk}}}{Area(S_\eta \bigcap P_\eta)}, \tag{5}$$

where S_η is the circle centered at η's current location, with a radius equal to η's social distance r_η; P_η is the path η is walking on; N_η^{walk} is the number of agents within the region $S_\eta \bigcap P_\eta$; $Area(S_\eta \bigcap P_\eta)$ returns the area of $S_\eta \bigcap P_\eta$. See Figure 4(a) for an illustration. η starts to perceive crowding if $\rho_\eta^{\text{walk}} > \rho_\eta^{\text{base}}$. We use ρ_η^{base} and ρ_η^{walk} in order to compute η's walking speed (Section 5.1.3).

Visiting Crowd Density. On the other hand, an agent perceives a *visiting crowd density* when it is visiting a site, defined as follows:

$$\rho_\eta^{\text{visit}} = \frac{N_\eta^{\text{visit}}}{Area(K_\eta)}, \tag{6}$$

where K_η is the site that η is visiting; N_η^{visit} is the number of other agents that are also visiting K_η currently; $Area(K_\eta)$ returns the area of site K_η. See Figure 4(b) for an illustration. η starts to perceive crowding if $\rho_\eta^{\text{visit}} > \rho_\eta^{\text{base}}$. In computing the coziness cost (Section 6.1), we use ρ_η^{base} and ρ_η^{visit}.

5.1.3 Locomotion

Default Walking Speed. Like a real human, each agent has an age, which determines its *default walking speed*. Studies [TranSafety, Inc 1997] found that the average walking speed of pedestrians aged 65 or above is 0.889 to $1.083\,\mathrm{m\,s^{-1}}$, while that of pedestrians aged below 65 is 1.042 to $1.508\,\mathrm{m\,s^{-1}}$. Accordingly, we define the default walking speed d_η for agent η as:

$$d_\eta = d_{\text{min}} + \xi_d(d_{\text{max}} - d_{\text{min}}), \tag{7}$$

where ξ_d is a uniformly distributed random number between 0 and 1; if agent η is aged 65 or above, $d_{\text{min}} = 0.889\,\mathrm{m\,s^{-1}}$ and $d_{\text{max}} = 1.083\,\mathrm{m\,s^{-1}}$, otherwise $d_{\text{min}} = 1.042\,\mathrm{m\,s^{-1}}$ and $d_{\text{max}} = 1.508\,\mathrm{m\,s^{-1}}$.

Actual Walking Speed. In general, people's walking speeds drop when the walking crowd density increases [Fang et al. 2003]. Accordingly, we define the *actual walking speed* v_η for agent η as:

$$v_\eta = \max(\frac{\rho_\eta^{\text{base}}}{\rho_\eta^{\text{walk}}} d_\eta, v_{min}), \tag{8}$$

where $v_{min} = 0.3\,\mathrm{m\,s^{-1}}$ is a lower-bound walking speed in a jam [Older 1964]. Essentially, v_η drops if more agents are present within η's social distance.

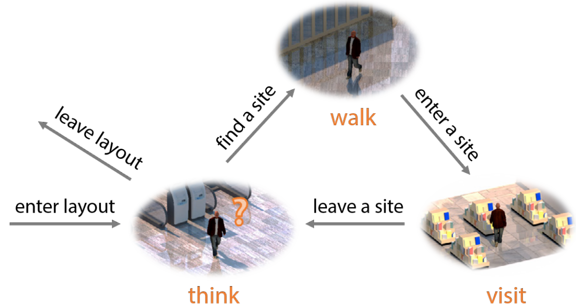


Figure 5: *State machine of an agent.*

5.2 Simulation Model

To evaluate a layout, we run an agent-based simulation using the agent model defined above. We keep track of agents' traversal experience, including their traversal paths, walking speeds, and perception of crowding. These will be used to compute the agent-based costs for evaluating the layout design.

More specifically, after building the graph of the layout and initializing the agents, we start our simulation, which progresses iteratively at a regular time interval (e.g., 1 second). At each iteration, it updates the status of each agent η according to the agent's state machine (depicted in Figure 5), as follows:

1. If the time reaches η's start time, η enters the layout from a randomly selected entrance. η's state changes to *think*.
2. If η's state is *think*, η determines the next site type to visit according to its visiting sequence. Then η finds a shortest path to a destination site of that site type. In case η has visited sites of all site types specified in its visiting sequence, or its visiting time is up, η finds the shortest path to the nearest entrance to leave the layout. In any case, η's state changes to *walk*.
3. If η's state is *walk*, η's position is updated according to its actual walking speed. In case η is planning to visit a site and it reaches the destination site, η's state changes to *visit*. In case η is planning to leave the layout and it reaches the destination entrance, it just leaves the layout.
4. If η's state is *visit*, η just stays in the site. When it has finished visiting, η leaves the site at a door node, and its state changes to *think*.

The simulation ends when all agents have left the layout.

6 Cost Functions

6.1 Agent-based Costs

An key component of our work is to optimize a layout with respect to agents' comfort and ease of movement. We achieve this by encoding these considerations in our agent-based costs, which can be evaluated quantitatively from an agent-based simulation. These agent-based costs are integrated into the total cost function (Equation (1)) to drive the optimization of a layout to become crowd-aware. We define three agent-based costs, namely, *Mobility*, *Accessibility* and *Coziness*, as depicted in Figure 6.

Mobility. A layout should be designed such that visitors can walk smoothly at their natural walking speed. A badly-designed layout, such as one with overly-narrow path, may frequently result in the formation of crowds which slow down people's movement.

We use the agents' walking speeds to evaluate a layout. According to our agent model (Section 5.1.3), an agent's default walking speed

Low Cost High Cost
(a) *Mobility*

Low Cost High Cost
(b) *Accessibility*

(c) *Coziness*

Figure 6: *Agent-based costs. (a) Mobility cost is low if agents walk smoothly at their default walking speed, and it is high if their movement is impeded by crowd; (b) Accessibility cost is low if agents only need to walk short distances to reach their destination sites, otherwise it is high; (c) Coziness cost is low if sites are not too crowded or too empty, otherwise it is high.*

d_η refers to the maximum walking speed it can attain in the absence of any crowd effect. An agent's *actual walking speed* v_η refers to the lowered speed in the presence of a crowd. To favor agents to walk at their *default walking speed* d_η, we define the mobility cost C_m as:

$$C_\mathrm{m}(\phi) = 1 - \frac{1}{N}\sum_i \frac{\bar{v}_i}{d_i}, \tag{9}$$

where N is the total number of agents; $\bar{v}_i$ is the average walking speed of the i^{th} agent throughout the simulation; d_i is its default walking speed. Note that $\bar{v}_i \leq d_i$.

Accessibility. Ideally, an agent should be able to find sites of all site types it plans to visit (specified in its *visiting sequence*) in the layout. Moreover, the agent does not have to walk a long distance to visit all these sites consecutively. To encode these considerations, we define the accessibility cost C_a as:

$$C_\mathrm{a}(\phi) = \frac{1}{NL}\sum_i \frac{1}{k_i}\sum_j l_{i,j}, \tag{10}$$

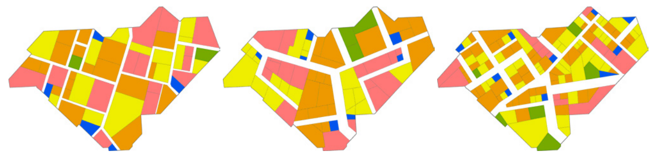
where N is the total number of agents; k_i is the number of sites that the i^th agent plans to visit; $l_{i,j}$ is the distance the i^th agent needs to walk from the $(j-1)^\mathrm{th}$ site (or an entrance if it just begins) to the j^th site (or an entrance if it is leaving) according to its *visiting sequence*; L is a normalization constant which is set to twice the contour length of the input layout domain. If the i^th agent cannot find a site of the j^th type it plans to visit, $l_{i,j}$ is set to L.

Coziness. People prefer to visit cozy places. The coziness of a layout is reflected by the visiting crowd density the agents perceive when visiting sites. Studies [Ng 2003] found that either an extremely low or high crowd density would degrade the perceived level of coziness. For example, a boutique which is too empty or too crowded would appear as uncozy. We define the coziness cost C_c as:

$$C_\mathrm{c}(\phi) = \frac{1}{N}\sum_i \frac{1}{q_i}\sum_j \Big[1 - \exp(-\frac{(\rho_{i,j}^{\mathrm{visit}} - \mu)^2}{2\sigma^2})\Big], \tag{11}$$

where N is the total number of agents; q_i is the total number of iterations that the i^th agent spends visiting a site; $\rho_{i,j}^{\mathrm{visit}}$ is the *visiting crowd density* the i^th agent experiences at the j^th iteration when it is visiting a site. We set $\mu = \rho_i^{\mathrm{base}}$ and $\sigma = \frac{\rho_i^{\mathrm{base}}}{2}$, where ρ_i^{base} is the i^th agent's *baseline crowd density*. Essentially, it penalizes if $\rho_{i,j}^{\mathrm{visit}}$ is either too low or too high with respect to ρ_i^{base}.

Figure 7 depicts the importance of the agent-based costs by showing the defective layouts synthesized with each of the agent-based costs being omitted.



(a) *No Mobility* **(b)** *No Accessibility* **(c)** *No Coziness*

Figure 7: *Layouts synthesized with one of the agent-based costs omitted. (a) Without the mobility cost term, some of the paths become undesirably narrow. (b) Without the accessibility cost, some of the sites are not conveniently connected by paths. (c) Without the coziness cost term, some of the sites are excessively small or big.*

6.2 Prior Costs

The prior costs in Equation (1) encode the design goals that influence the layout to be synthesized. We consider three prior costs: the prior total number of sites; the prior number of each type of sites; and the prior ratio of sites' area to layout area (commonly known as floor area ratio). These prior values can be learned from training layout data, or manually specified by users depending on the requirements for the type of layouts to be synthesized. Taking the prior total number of sites as an example, a regional mall (with size from 40 to 100 acres) usually consists of 40 to 80 stores while a neighborhood shopping center (with size from 3 to 5 acres) consists of 5 to 20 tenants [ICSC 2015]. Further details of the prior costs can be found in our supplementary material.

7 Approximation by Regression

Theoretically, we could run an agent-based simulation for each evaluation of the agent-based costs during the layout optimization. However, such an approach is too costly to be practical. A simulation of 1000 agents navigating in a layout for 2 simulated hours with a regular time step of 1 second would take 2 to 5 minutes, corresponding to about 7.2 million agent updates in total. Moreover, optimizing a layout requires hundreds of iterations of such evaluations, which would take about 10 to 25 hours in total. We include further analysis about the choice of time step and reliability of the simulation data to justify the use of nonlinear approximation in our supplementary material.

We thus learn the relationship between layout features and the agent-based costs obtained from a full simulation through nonlinear regression. Then we can use the trained regressors to predict (approximate) the agent-based costs for a new layout, without the need of a full-blown simulation. Our trained regressors can predict the agent-based costs reasonably well in about 0.5 to 2 seconds, compared to 2 to 5 minutes needed for running one full simulation, making them practical to be used for layout optimization. We note that AlHalawani et al. [2014] trained classifiers to distinguish city styles from layout features, while Vanegas et al. [2012a] used neural networks to predict how well a city generated by a certain set of procedural rules and their parameters would satisfy some macroscopic requirements such as ensuring enough sunlight exposure. We in-

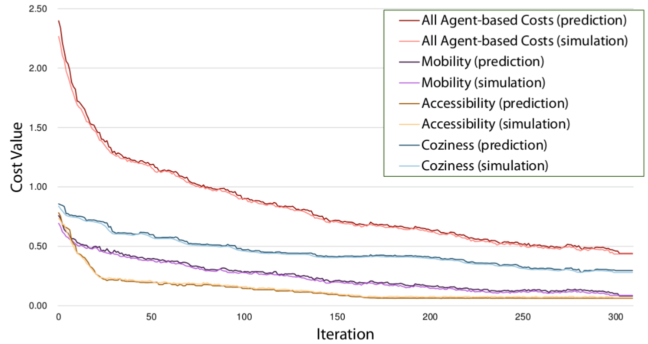


Figure 8: *Comparing the costs computed by full simulation and approximated by our regressors, for layouts synthesized throughout the optimization of our illustrative shopping mall example.*

stead train regressors to capture the relationship between the agent-based metrics, and the geometrical and topological features of a layout, in order to synthesize layouts without running full-blown crowd simulations.

7.1 Layout Features

We select various features representing the geometric, topological, and general characteristics of a layout for the nonlinear regression. The geometric features include statistics related to the path length, path width and the site area. The topological features include statistics related to the node and edge valence of the graph, and the travel distance. The general features include statistics related to the site type, the path-to-site ratio, centrality, and so on. The definitions of the layout features can be found in our supplementary material.

7.2 Training

To prepare our training data, we collected about 20 layouts for each category (shopping malls, theme parks, train station buildings, campuses). In total, we collected about 80 layouts from different parts of the world and included both positive and negative examples, as evaluated by AnyLogic [2016] and PathFinder [2016]. We used ArcGIS [ESRI 2016], a geographical information systems software, to digitize the collected layouts into standard spatial data format.

For each layout, we ran agent-based simulations using different numbers of agents (1000, 2000 and 3000) for different numbers of simulated hours (1 hour and 2 hours). Altogether we have 6 combinations of settings for each layout, giving us 480 training samples in total. For each training sample, we extracted the layout features (Section 7.1) and computed the agent-based costs (Section 6.1) to train our regressors.

We choose Random Forests, an ensemble learning method known to be robust to outliers and overfitting, for our nonlinear regression task [Liaw and Wiener 2002]. Our experiments reveal that Random Forests regressors achieve better accuracy than alternative methods, such as AdaBoost. For more details of the random forests regressors, please refer to our supplementary material.

7.3 Prediction

We use the regressors trained offline for runtime optimization. That is, in each iteration of the optimization, instead of running a full simulation to compute the agent-based costs, we extract the layout features from the current layout, and feed them into the trained regressors to obtain the approximated agent-based costs, $\hat{C}_{\mathrm{m}}$, $\hat{C}_{\mathrm{a}}$ and $\hat{C}_{\mathrm{c}}$ (Equation (2)). Figure 8 and Table 3 show the trends and prediction accuracies of different agent-based costs throughout the optimization of the shopping mall layout used for illustration (this layout is not used for training). The overall accuracy is about 95.24%.

	Mobility	Accessibility	Coziness	All Agent-based Costs
Accuracy	91.78%	89.98%	97.09%	95.24%

Table 3: *Prediction accuracy by random forests regressors.*

Cost	Feature	Importance
Mobility	Edge Width	21.51%
	Travel Distance	16.19%
	Betweenness Centrality	15.28%
Accessibility	Travel Convenience	28.57%
	Node Valence	15.76%
	Closeness Centrality	14.65%
Coziness	Site Area	26.03%
	Path Area Ratio	14.66%
	Site Histogram	9.16%

Table 4: *Important features for predicting each agent-based cost found in training our random forests regressors. The top three important features are listed. Each percentage refers to the degree by which the prediction of each cost relies on each feature.*

Table 4 shows the top three important features for predicting each agent-based cost found in training our random forests regressors. *Edge Width* affects the walking crowd density and hence the walking speed of agents, and is highly related to the prediction of the mobility cost. *Travel Convenience* encodes the distance between any two sites, and is highly related to the prediction of the accessibility cost. *Site Area* affects the visiting crowd density perceived by agents, and is highly related to the prediction of the coziness cost.

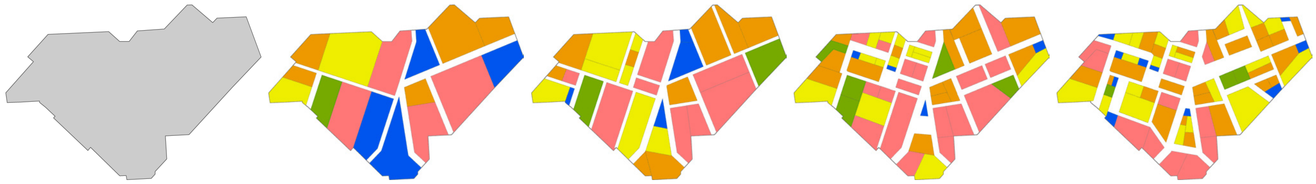
8 Layout Optimization

We synthesize crowd-aware layouts by optimizing against the approximated total cost function (Equation (2)). The agent-based considerations ensure that the synthesized layouts are optimal with respect to comfort and ease of movement. This optimization problem has a highly complex solution space with multiple local minima, mainly due to the interdependency of the paths and sites, and their non-convex relationships with the agent-based costs. Thus it is difficult to obtain a closed-form solution, if possible at all.

We therefore use a stochastic optimization technique, namely, simulated annealing [Kirkpatrick et al. 1983] with a Metropolis-Hastings [Metropolis et al. 1953; Hastings 1970] state-searching step to search for an optimal layout, akin to other recent layout synthesis works [Merrell et al. 2010; Vanegas et al. 2012a; Yu et al. 2012; Bao et al. 2013]. Due to its stochastic nature, the technique can be applied multiple times to synthesize different optimal layouts. We refer the reader to our supplementary material and the literature for further details of simulated annealing.

Figure 9 shows the layouts of the example shopping mall synthesized at different iterations of the optimization process. We terminate the optimization when the percentage decrease in the approximated total cost $\hat{C}(\phi)$ is lower than a small threshold. The optimization typically requires 300 to 400 iterations. The whole optimization takes about 2 to 5 minutes using regressor approximation, compared with 10 to 25 hours using full simulations.

Figure 10 compares layouts synthesized by optimization using full simulation (Equation (1)) and using the regressors (Equation (2)). For each of the final layouts, we also run a full simulation to compute the total cost, to verify that the total cost does decrease to a small value. The comparison shows that the layouts synthesized using the regressors closely resemble those synthesized using full simulation, qualitatively and quantitatively.



(a) *Input Layout Domain* — $C(\phi) = 2.36$ **(b)** 50 *Iterations* — $C(\phi) = 1.68$ **(c)** 100 *Iterations* — $C(\phi) = 1.35$ **(d)** 150 *Iterations* — $C(\phi) = 0.92$ **(e)** 320 *Iterations (result)*

Figure 9: *Layouts synthesized at different iterations by optimizing against the approximated total cost, $\hat{C}(\phi)$. At each iteration shown, we also show the original total cost $C(\phi)$ computed by a full simulation.*

8.1 Proposal Moves

We devise a set of proposal moves $\phi \rightarrow \phi'$ to alter the layout and explore the solution space effectively throughout the optimization. Figure 11 depicts these moves. Some moves alter the layout significantly, allowing our optimizer to rapidly explore the space of possible layouts and escape local minima, while other moves alter the layout slightly, allowing our optimizer to gradually refine the solution.

Sliding Boundaries. The common boundary between two or more sites is slid to change the shapes of the sites slightly.

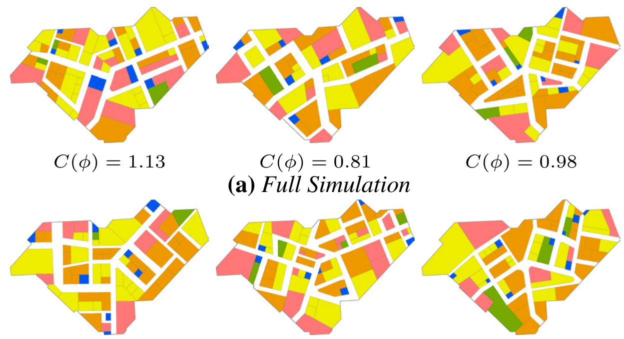
Changing Path Widths. A randomly-selected path is narrowed or widened. This operation also changes the shapes of the sites adjacent to the path.

Changing Site Types. A site is randomly selected and its type is changed.

Adding and Removing Paths. Adding a path refers to turning the common boundary between two or more sites into a new path. Removing a path refers to turning a path between two or more sites into their common boundary. A connectivity check is done before removing any path, to ensure that all sites still remain reachable after removing the path.

Merging and Splitting Sites. Existing sites are merged into bigger sites, or split into smaller sites.

Swapping Blocks. A block is a group of sites surrounded by paths. Two randomly selected blocks exchange the number of sites and type of sites within, and go through a series of splitting and assignment to redesign the layout at the block level. Essentially the sites within the blocks are swapped, and their shapes can change too. This move usually leads to a significant change in the total cost, and can help our optimizer jump out of a local minimum.



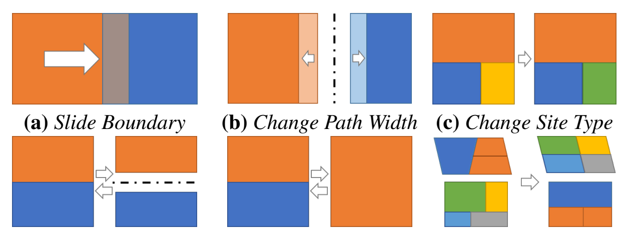
$C(\phi) = 1.13$ $C(\phi) = 0.81$ $C(\phi) = 0.98$

(a) *Full Simulation*

$C(\phi) = 0.78$ $C(\phi) = 0.98$ $C(\phi) = 0.88$

(b) *Approximation by Regressors*

Figure 10: *Layouts synthesized by optimizing against the agent-based costs (a) computed by full simulation and (b) approximated by regressors. For each layout shown, we also run a full simulation to show the original total cost, $C(\phi)$. The initial layouts from which these layouts are synthesized have $C(\phi) = 4.30$ to $C(\phi) = 5.00$.*



(a) *Slide Boundary* **(b)** *Change Path Width* **(c)** *Change Site Type*

(d) *Add & Remove Path* **(e)** *Merge & Split Sites* **(f)** *Swap Blocks*

Figure 11: *Different proposal moves.*

9 Results

We implemented our framework as an interactive layout modeling tool using C# and the .NET Framework. For the training part, we used the random forests regressor provided by scikit-learn [INRIA 2016] in Python. We measure the performance of our prototype on an Intel Core i7-3930K machine running at 3.2 GHz with 8 GB of RAM. In our simulation, we set the agents' *total length of visits* to 1.5 simulated hours unless otherwise specified. All agents enter the layout in the first 0.5 hour, and our simulation runs for 2 simulated hours.

Different Layouts. We have synthesized layouts for different types of mid-scale structures: *Mall 1*, *Mall 2*, *Theme Park*, *Train Station Building* and *Campus*. The original layouts are real world layouts in North America and Asia. *Theme Park* refers to the layout of Disneyland in Tokyo. We set the number of agents to 1000 for *Mall 2*, 2000 for *Mall 1* and *Campus*, and 3000 for *Train Station Building* and *Theme Park*. Figure 12 shows the layouts synthesized by our system using the boundaries of real world layouts. We include the training details and the computation time needed for all the syntheses in our supplementary material.

For *Mall 1*, we can synthesize sites of different types around the atrium. The atrium, which serves as a place for exhibitions and entertainment activities, is treated as a hard constraint. Sites of different types are synthesized around it. Our accessibility cost helps distribute the facilities evenly throughout a layout, to ensure easy access from any site; it also encourages the addition of small paths (shortcuts) throughout a layout, to provide easy access between sites, and avoid crowding and congestion.

For *Mall 2*, the original mall shows a poor design with narrow paths, cramped stores and an inconvenient distribution of shops and facilities. Our system synthesizes layouts with wider paths and reasonably-distributed sites (especially for Facilities, and Electronics & IT). Our supplementary material includes the heat map visualizations of a remodeled layout, which show improved agent-based costs compared to the original layout.

For *Train Station Building*, we set an open space near the middle by assigning a fixed rectangular atrium (for Synthesis 1) or circular atrium (for Synthesis 2). Our syntheses usually form an interesting

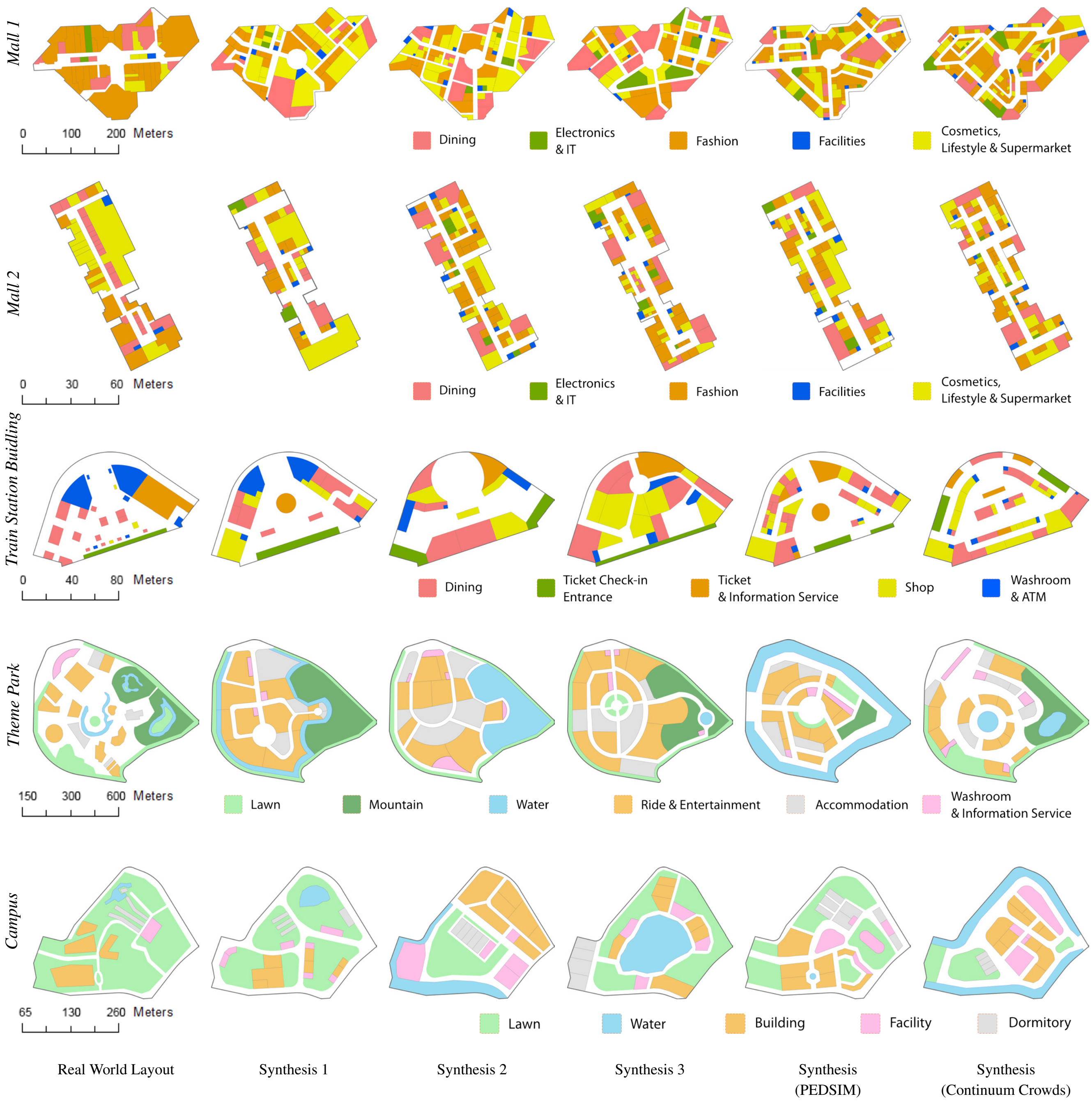


Figure 12: *Different types of crowd-aware layouts synthesized by our approach.*

symmetric pattern around the open space. We lengthen the $think$ phase of the agents' state machine by 20 to 30 simulated seconds, to mimic the delay in decision-making due to uncertainty commonly encountered in a train station. In all syntheses, as all agents need to visit the Ticket & Information Service site at some point (which is specified in their *visiting sequences*), our optimizer either places the Ticket & Information Service site in the middle (in Synthesis 1) to make it easily accessible in accordance with the accessibility cost; or sets it big (in Synthesis 2 and 3) so that it can accommodate more agents simultaneously in accordance with the coziness cost.

For *Theme Park*, akin to the real world layout, we allow the addition of curved paths and circular sites, which our proposal moves in Section 8.1 can handle as well. However, our proposal moves cannot generate the sites with complex shapes observed in the real world layout, such as the Water sites around the center lawn and inside the mountains. In Synthesis 1, we set the river surrounding the layout as a hard constraint. Our system is able to synthesize sites representing lawn, mountain, accommodation, and so on, to populate the layouts. We set the agents to behave like theme-park visitors: most of them want to visit Ride & Entertainment sites multiple times. Accordingly, in each layout our optimizer places multiple Ride & Entertainment sites, which are connected by circular paths to ensure convenient access from one another.

For *Campus*, as students who live in the dormitory need to visit

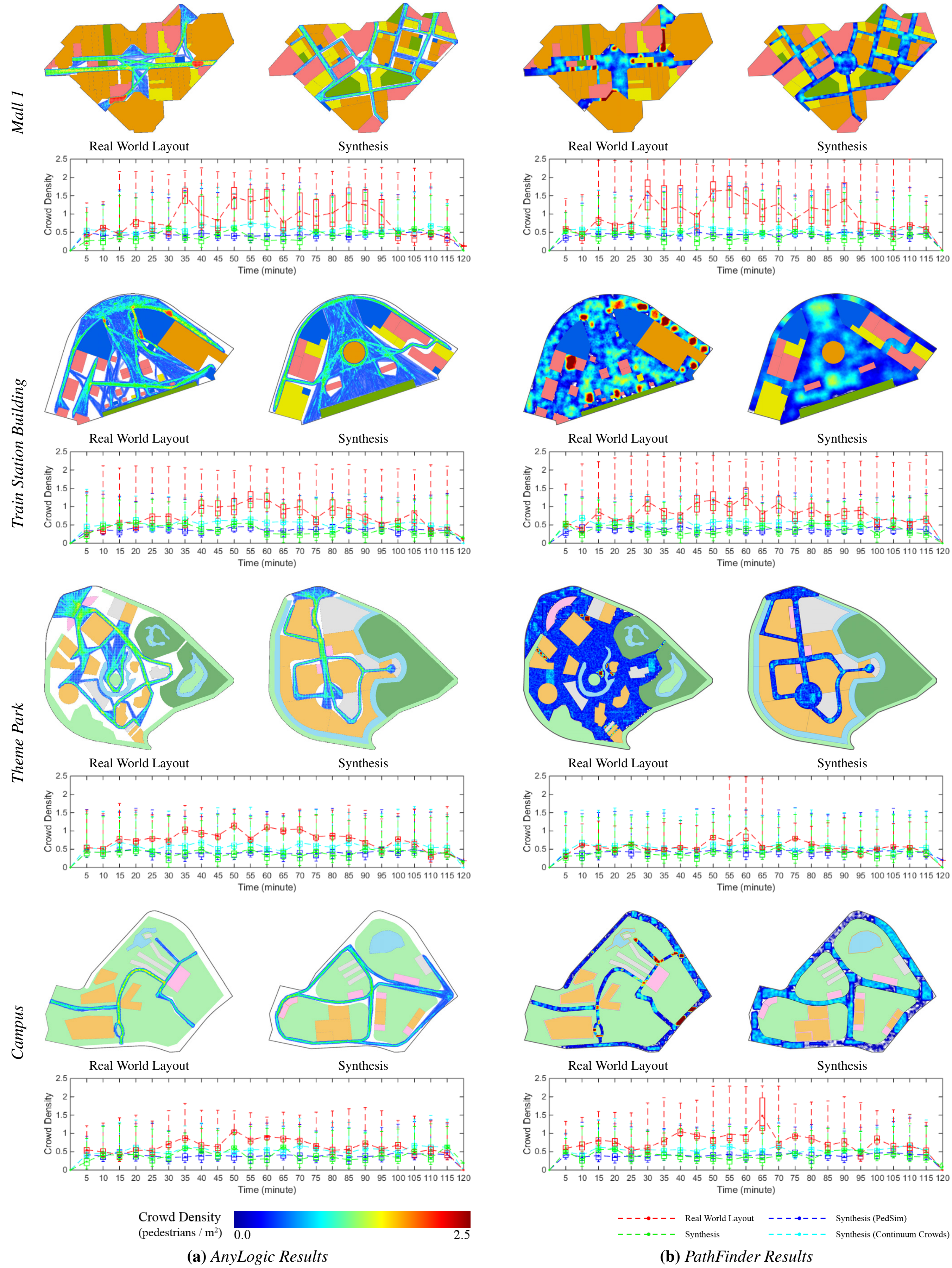


Figure 13: *Evaluating real world layouts and our syntheses by (a) AnyLogic and (b) PathFinder. Heat maps show crowd densities at the middle of the simulation. Curves and box plots show the means and statistics of crowd densities versus simulation time. Please refer to our supplementary material for the evaluation of Mall 2 and the corresponding syntheses.*

	Mall 1	Mall 2	Train Station	Theme Park	Campus
Real World Layout	0.9348	0.8391	0.6908	0.5306	0.6074
Synthesis (Our Simulation Model)	0.5913	**0.4086**	**0.4260**	**0.4839**	0.5382
Synthesis (PEDSIM)	**0.5608**	0.5096	0.4695	0.5039	**0.4721**
Synthesis (Continuum Crowds)	0.6958	0.6612	0.5395	0.5148	0.5411
User Syntheses (Average)	0.5924	0.4428	-	-	-

(a) *AnyLogic Results*

	Mall 1	Mall 2	Train Station	Theme Park	Campus
Real World Layout	0.9069	0.9130	0.8917	0.4891	0.5913
Synthesis (Our Simulation Model)	0.5304	**0.4478**	**0.4956**	0.4882	0.4826
Synthesis (PEDSIM)	**0.5221**	0.5096	0.5173	**0.4572**	**0.4673**
Synthesis (Continuum Crowds)	0.6847	0.6298	0.5654	0.4967	0.5138
User Syntheses (Average)	0.5893	0.4581	-	-	-

(b) *PathFinder Results*

Table 5: *Average crowd densities of layouts in evaluation. The lowest crowd density is depicted in bold.*

the facilities (e.g., canteen, student services, recreation) frequently for different daily needs and activities, the Facility sites are always placed close to the Dormitory sites to ensure convenient access.

Alternative Simulation Models. To verify the effectiveness and generality of our approach for synthesizing crowd-aware layouts by nonlinear regression, we replace our agent-based simulation model with two popular crowd simulation models, *PEDSIM* [Gloor 2016] and *Continuum Crowds* [Treuille et al. 2006].

PEDSIM is a microscopic pedestrian crowd simulation framework. The framework is based on the *social force model* [Helbing and Molnar 1995], which controls the direction and velocity of agents, as well as the interaction of agents with the environment and with each other. *Continuum Crowds* is based on continuum dynamics, where a dynamic potential field simultaneously integrates global navigation with moving obstacles such as other agents. The model assumes that agents choose the minimum distance paths to their destinations while avoiding congestion and other time-consuming situations. In addition, agents prefer to minimize their exposure to "discomfort fields". The routes of agents are determined by minimizing their walking distances, time and perceived discomfort. Both models are microscopic and considerably more complicated than our simulation model used for illustrating our approach.

We trained regressors based on crowd simulations obtained from running each of these alternative simulation models, using the same speed limits for the agents as described in Section 5. The *social distances* in our model are used to set the radii of *private spheres* in *PEDSIM*, and the radii of *boundary discs* in *Continuum Crowds*.

We then optimize layouts with respect to the agent-based costs approximated by the regressors obtained from each of the alternative simulation models. The fifth and sixth columns of Figure 12 show the synthesized layouts, which look similar to those synthesized by our simulation model. We verify in our evaluation that these synthesized layouts also exhibit improved crowd flow properties.

Evaluation. We evaluate the real world and our synthesized layouts with two third-party crowd simulation software, *AnyLogic* [2016] and *PathFinder* [2016]. By setting the same parameters (number of agents, simulation time, agents' visiting sequences) as in our simulation, we populate agents to navigate in the layouts. As computing crowd densities is supported by both software, we select it as our metric for evaluating the layouts.

Figure 13 shows the evaluation results. Each evaluation runs for two simulated hours. The heat maps visualize the crowd densities at the 60$^{\text{th}}$ simulated minute. Box plots and curves show respectively the statistics (minimum, maximum, the first and third quartiles) and means of crowd densities over the two simulated hours. Please refer to our supplementary material for the evaluation of Mall 2 and the corresponding syntheses. Table 5 lists the average crowd density over the two simulated hours.

(a) *Original* **(b)** *Grazers-friendly* **(c)** *Flea Market*

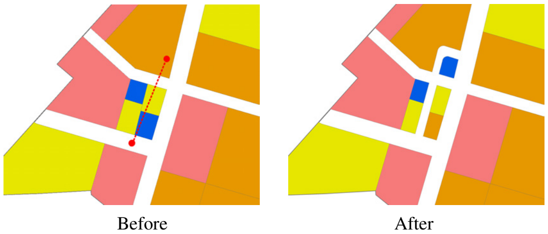
Figure 14: *Changing agents' properties and the resulting layouts. (a) Original layout. (b) Increasing the proportion of 'grazer' agents, the synthesized layout shows more dining places (pink) throughout the layout. (c) Raising the agents' tolerance for coziness and shortening their social distance, the synthesized layout mimics a flea market populated with small shops and connected by narrow paths.*

For indoor environments (*Mall 1*, *Mall 2* and *Train Station Building*), our synthesized layouts show significant improvement in crowd densities compared to the real world layouts, as shown in the plots, the heat map visualization and the table. However, for outdoor environments (*Theme Park* and *Campus*), the improvement brought about by our synthesized layouts is less prominent. One possible explanation is that outdoor layouts are usually designed with a much bigger capacity than the maximal number of agents (3000) we have tested. This results in more open space and wider pathways in real world outdoor layouts than in our syntheses (see Figure 12). Nevertheless, we find that the layouts synthesized by using our simulation model, *PEDSIM* and *Continuum Crowds* result in similar crowd densities, regardless of the layout type.

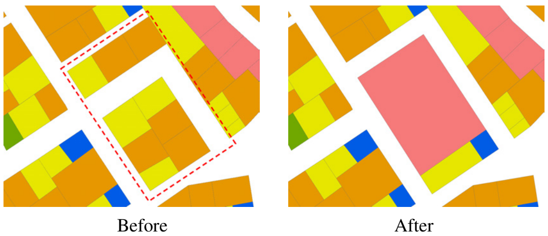
Changing Agents' Properties. Using our approach, a user can easily change agents' properties and obtain optimized layouts accordingly. This makes it intuitive for users to design layouts based on visitors' statistics and habits, which can be obtained by surveys. Our optimizer automatically takes into account the visiting habits and synthesizes an appropriate layout. Figure 14(b) depicts a mall layout synthesized with more *grazers*, which shows more dining places throughout the space. Figure 14(c) shows another example in which agents are given a high tolerance for coziness and a short social distance, to mimic shoppers in a flea market. The synthesized layout shows small shops and narrow paths as a result, which are common in a flea market.

User Interaction. The speedup provided by our regressor approximation enables layout modeling at interactive rates—an advantage that optimization by full simulation cannot afford due to its costly computations. Figure 15 shows two examples of adding a path and merging a group of sites into a block. In both cases, our optimizer refines the modified layout automatically such that it remains optimal with respect to different agent-based criteria. As the modification is usually small and local, the refinement is much easier than an entire optimization for layout synthesis and can usually finish in several seconds. Our interface quickly displays the refined layouts to the user for further editing. Please refer to our supplementary video for interactive demos.

User Synthesis. We invited 22 participants to design layouts using our research prototype. The participants were senior undergraduate students in architectural design, who by training had experience in using commercial architectural design software. After a short tutorial of 10 minutes about our user interface, all participants were given the real world layouts of *Mall 1* and *Mall 2*. They were asked to synthesize new layouts by changing the parameters (e.g., agent properties) in the tool and triggering optimization. The participants were free to modify the synthesized layouts obtained from the optimization with basic operations same as the proposal moves (Section 8.1), and with more advanced operations depicted in Figure 15. They were allowed to modify their layouts until they felt satisfied that their layouts would be comfortable for visitors to navigate in. On average, it took 7 minutes and 23 seconds for the participants to



(a) *Adding Paths*



(b) *Merging Sites into a Block*

Figure 15: *User interaction. (a) The user drags a line over the sites around which to add paths. Our tool automatically adds paths around the sites, with path width and turning angles optimal to agents' movement. (b) The user draws a box to enclose the sites to be merged into a block. The new block is then automatically split into smaller sites which exhibit a lower coziness cost than a huge site does. In both operations, our optimizer automatically assigns the types of the new sites considering the existing sites. For example, in (b), the newly-formed big site is assigned as a dining place (pink) because there is no easily-accessible dining place nearby.*

finish designing a layout.

All the layouts synthesized by the participants are included in our supplementary material, together with their crowd densities obtained from running *AnyLogic* and *PathFinder* using the same settings as in our evaluation. Table 5 shows the average crowd densities of these layouts. For *Mall 1*, the average crowd densities are 0.5924 (*AnyLogic*) and 0.5893 (*PathFinder*) pedestrians per square meter. For *Mall 2*, the average crowd densities are 0.4428 (*AnyLogic*) and 0.4581 (*PathFinder*) pedestrians per square meter.

We received useful feedback from the participants about our prototype. Some participants suggested that our interface could support multi-layer layout design so that they could design their layouts hierarchically. Most participants find our tool useful to augment existing commercial architectural design software (e.g., Rhinoceros 3D, Autodesk Revit) to incorporate crowd flow considerations into layout design. They expressed that even though wider paths and bigger sites may intuitively lower the chance of congestion, it is rather hard to precisely adjust all the paths and sites manually. In addition, there are often space constraints and aesthetic factors to consider in adjusting the paths and sites, making the adjustment rather non-trivial in practice. It is particularly difficult for participants to estimate the crowd flow effects a layout design would induce, and it is helpful that our optimization considers such effects automatically in layout synthesis and during user interaction for refining a layout.

In addition, we consulted three professors in architecture about our research prototype. We received positive comments regarding the functionality of our tool in visualizing iterative refinement with respect to crowd properties; the automatic and efficient consideration of crowd properties in layout design; and the interactivity for exploring different solutions.

Limitations. We have designed a rather simple agent model to illustrate the capability of the overall framework. However, this model can be easily extended to more faithfully reflect human behaviors. For example, the fixed length of visit can be replaced by probability distributions; more age groups can be incorporated to calculate walking speed. In addition to crowd properties related to movements, a people-oriented layout design should also consider visibility, emotion and social interactions, and grouping behaviors. Figure 12 (*Train Station Building*, Synthesis 3) shows a failure case of the current system. The ticket machines are visually blocked from the ticket check-in entrance by other sites such as shops, despite that the ticket machines are indeed connected to the entrance by paths. Such layout can be confusing because passengers may not know where to go after buying tickets, if the check-in entrance is not visible. This problem can be addressed by incorporating visibility into the agent model and agent-based costs. Mall visitors may also get entertained or distracted by live activities or coincident events. Groups of people such as friends and families usually navigate together. It is interesting to extend our agent model to handle such social interactions and grouping behaviors in the future.

Nevertheless, our framework is not constrained by the current agent model, and more sophisticated agent models and simulation methods can be plugged in with ease, as illustrated in our verifications using different simulation methods and software. Most recently, data-driven methods [Guy et al. 2012; Wilkie et al. 2013] have pushed the realism of traditional crowd simulation to a whole new level. These works can be incorporated to further improve the fidelity of our results.

10 Discussion

We have demonstrated a layout design framework which automatically incorporates crowd flow considerations into the design loop, for synthesizing layouts, changing layout styles, and remodeling and improving existing layouts. While it could be intuitive for architects, property managers or community planners to specify knowledge of human movement such as walking speeds and navigation paths, how such knowledge could be incorporated in guiding the design of a layout is non-trivial and challenging, especially for mid-scale environments that involve thousands of visitors each having different physical and behavioral characteristics. The key insights of our approach include the evaluation of three crowd flow properties: mobility, accessibility and coziness, using agent-based simulation models; the integration of such evaluation in layout optimization; and the approximation of full simulation with nonlinear regression. In the end, we have achieved a practical system for generating *crowd-aware mid-scale layouts*.

There are a few open problems remaining for future exploration. We have so far only investigated crowd flow properties in designing the layouts. In practice, there are many other factors that may influence the design. For example, a property manager may wish to maximize the traffic and profit of a shopping mall and is less concerned with occasional congestions. As a result, luxury brands are usually located at the center while restrooms at corners. Safety is also a critical consideration in layout design. Regardless of the category of a building, the layout should ensure that pedestrians can quickly evacuate in case of fire or emergency. Another design philosophy that we have not explored in this work is the clustering of similar sites. A customer may want to visit multiple fashion stores in a row to find her dream dress, or compare the price of some electronics sold in different stores. To facilitate such shopping behaviors, the designer will need to trade off between the general breath-first shoppers and such goal-oriented shoppers.

It is also interesting to study how different layout and architectural features relate to functionality, aesthetics, psychology and even sociology. We believe that such relationships do exist and can be computationally modeled, as architects and urban designers do create their designs with such considerations in mind. In addition, we are interested in applying recent advances in 3D shape synthesis [Xu et al. 2012] for generating a more diverse set of layouts.

Acknowledgements

We are grateful to the anonymous reviewers for their constructive comments. We also thank Michael S. Brown for narrating the video; Zhipeng Mo and Jay Daligdig for providing help on evaluation and teaser; Keng Hua Chong, Sawako Kaijima and Bige Tunçer for consultation on user experience of our research prototype; Roland Bouffanais and Zheng Cui for providing advice on crowd simulation techniques; and NVIDIA Corporation for the graphics card donation. Sai-Kit Yeung is supported by Singapore MOE Academic Research Fund MOE2013-T2-1-159 and SUTD-MIT International Design Center Grant IDG31300106. We acknowledge the support of the SUTD Digital Manufacturing and Design (DManD) Centre which is supported by the National Research Foundation (NRF) of Singapore. This research is also supported by the National Research Foundation, Prime Minister's Office, Singapore under its IDM Futures Funding Initiative. Part of the work was done when Tian was visiting UMass Boston. Lap-Fai Yu is supported by the UMass Boston StartUp Grant P20150000029280, and the Joseph P. Healey Research Grant provided by the Office of the President, and the Office of the Vice Provost for Research and Strategic Initiatives & Dean of Graduate Studies of UMass Boston. Kun Zhou is supported by the National Natural Science Foundation (NSF) of China (No. 61272305).

References

ALHALAWANI, S., AND MITRA, N. J. 2015. Congestion-aware warehouse flow analysis and optimization. In *Advances in Visual Computing*. Springer, 702–711.

ALHALAWANI, S., YANG, Y.-L., WONKA, P., AND MITRA, N. J. 2014. What makes London work like London? *Computer Graphics Forum 33*, 5 (Aug.), 157–165.

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.

ALIAGA, D. G., VANEGAS, C. A., AND BENEŠ, B. 2008. Interactive example-based urban layout synthesis. *ACM Trans. Graph. 27*, 5 (Dec.), 160:1–160:10.

ALIAGA, D. G. 2012. 3d design and modeling of smart cities from a computer graphics perspective. *International Scholarly Research Notices 2012*.

ASCHWANDEN, G. D., HAEGLER, S., BOSCHÉ, F. N., VAN GOOL, L., AND SCHMITT, G. 2011. Empiric design evaluation in urban planning. *Automation in construction 20*, 3 (May), 299–310.

BAO, F., YAN, D.-M., MITRA, N. J., AND WONKA, P. 2013. Generating and exploring good building layouts. *ACM Trans. Graph. 32*, 4 (July), 122:1–122:10.

BENEŠ, J., WILKIE, A., AND KŘIVÁNEK, J. 2014. Procedural modelling of urban road networks. *Computer Graphics Forum 33*, 6, 132–142.

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

BROWN, M. G. 1999. Design and value: Spatial form and the economic failure of a mall. *Journal of Real Estate Research 17*, 189–225.

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.

CHEN, G., ESCH, G., WONKA, P., MÜLLER, P., AND ZHANG, E. 2008. Interactive procedural street modeling. *ACM Trans. Graph. 27*, 3 (Aug.), 103:1–103:10.

CLAVE, S. A. 2007. *The Global Theme Park Industry*. CABI.

D. BEYARD, M., CORRIGAN, M. B., KRAMER, A., PAWLUKIEWICZ, M., AND BACH, A. 2006. *Ten Principles of Rethinking the Mall*. Urban Land Institute.

ESRI, 2016. ArcGIS. http://arcgis.com. Accessed: 01-01-2016.

FANG, Z., LO, S., AND LU, J. 2003. On the relationship between crowd density and movement velocity. *Fire Safety Journal 38*, 3, 271–283.

FISHER, M., SAVVA, M., LI, Y., HANRAHAN, P., AND NIESSNER, M. 2015. Activity-centric scene synthesis for functional 3d scene modeling. *ACM Trans. Graph. 34*, 6 (Oct.), 179:1–179:13.

GALIN, E., PEYTAVIE, A., MARÉCHAL, N., AND GUÉRIN, E. 2010. Procedural generation of roads. *Computer Graphics Forum 29*, 2, 429–438.

GALIN, E., PEYTAVIE, A., GUÉRIN, E., AND BENES, B. 2011. Authoring hierarchical road networks. *Computer Graphics Forum 30*, 7, 2021–2030.

GALLE, P. 1981. An algorithm for exhaustive generation of building floor plans. *Communications of ACM 24*, 12 (Dec.), 813–825.

GLOOR, C., 2016. Pedsim. http://pedsim.silmaril.org. Accessed: 01-01-2016.

GUY, S. J., VAN DEN BERG, J., LIU, W., LAU, R., LIN, M. C., AND MANOCHA, D. 2012. A statistical similarity measure for aggregate crowd dynamics. *ACM Trans. Graph. 31*, 6 (Nov.), 190:1–190:11.

HALL, E. T. 1990. *The Hidden Dimension*. A Doubleday anchor book. Anchor Books.

HARADA, M., WITKIN, A., AND BARAFF, D. 1995. Interactive physically-based manipulation of discrete/continuous models. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, SIGGRAPH '95, 199–208.

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

HELBING, D., AND MOLNAR, P. 1995. Social force model for pedestrian dynamics. *Physical review E 51*, 5, 4282.

HUERRE, S., LEE, J., LIN, M., AND O'SULLIVAN, C. 2010. Simulating believable crowd and group behaviors. In *SIGGRAPH Asia 2010 Courses*.

ICSC. 2015. *U.S. Shopping-Center Classification and Characteristics*. ICSC Research and CoStar Realty Information, Inc., mar.

INRIA, 2016. scikit-learn. http://scikit-learn.org. Accessed: 01-01-2016.

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

LI, X., AND LIU, X. 2008. Embedding sustainable development strategies in agent-based models for use as a planning tool. *International Journal of Geographical Information Science 22*, 1, 21–45.

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.

LIU, H., YANG, Y.-L., ALHALAWANI, S., AND MITRA, N. J. 2013. Constraint-aware interior layout exploration for precast concrete-based buildings. *Visual Computer (CGI Special Issue)*.

MACHLEIT, K. A., EROGLU, S. A., AND MANTEL, S. P. 2000. Perceived retail crowding and shopping satisfaction: What modifies this relationship? *Journal of Consumer Psychology 9*, 1, 29–42.

MASSIVE SOFTWARE, 2016. Massive. http://massivesoftware.com. Accessed: 01-01-2016.

MERRELL, P., SCHKUFZA, E., AND KOLTUN, V. 2010. Computer-generated residential building layouts. *ACM Trans. Graph. 29*, 6 (Dec.), 181:1–181:12.

MERRELL, P., SCHKUFZA, E., LI, Z., AGRAWALA, M., AND KOLTUN, V. 2011. Interactive furniture layout using interior design guidelines. *ACM Trans. Graph. 30*, 4 (July), 87:1–87:10.

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.

MICHALEK, J., CHOUDHARY, R., AND PAPALAMBROS, P. 2002. Architectural layout design optimization. *Engineering Optimization 34*, 12, 461–484.

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.

NILES, R., 2012. Robert's rules for the perfect theme park. http://themeparkinsider.com/flume/201212/3304, Dec. Accessed: 01-01-2016.

NISHIDA, G., GARCIA-DORADO, I., AND ALIAGA, D. G. 2015. Example-driven procedural urban roads. *Computer Graphics Forum*.

OLDER, S. J. 1964. *Pedestrians*. Department of Scientific and Industrial Research Laboratory, Crowthorne, England.

PARISH, Y. I. H., AND MÜLLER, P. 2001. Procedural modeling of cities. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, SIGGRAPH '01, 301–308.

PENG, C.-H., YANG, Y.-L., AND WONKA, P. 2014. Computing layouts with deformable templates. *ACM Trans. Graph. 33*, 4 (July), 99:1–99:11.

PENG, C.-H., MITRA, N. J., BAO, F., YAN, D.-M., AND WONKA, P. 2015. Computational Network Design from Functional Specifications. *ArXiv e-prints* (Oct.).

PETTRÉ, J., KALLMANN, M., AND LIN, M. C. 2008. Motion planning and autonomy for virtual humans. In *ACM SIGGRAPH 2008 Courses*, ACM, 42.

PRUSINKIEWICZ, P., AND LINDENMAYER, A. 1996. *The Algorithmic Beauty of Plants*. Springer-Verlag New York, Inc., New York, NY, USA.

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.

SHAVIV, E. 1974. *A Model for Space Allocation in Complex Buildings: A Computer Graphics Approach*. Working papers. Faculty of Architecture and Town Planning.

SUN, J., YU, X., BACIU, G., AND GREEN, M. 2002. Template-based generation of road networks for virtual city modeling. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, ACM, New York, NY, USA, VRST '02, 33–40.

THE ANYLOGIC COMPANY, 2016. Anylogic. http://anylogic.com. Accessed: 01-01-2016.

THUNDERHEAD ENGINEERING CONSULTANTS, INC., 2016. Pathfinder. http://thunderheadeng.com/pathfinder. Accessed: 01-01-2016.

TRANSAFETY, INC. 1997. Study compares older and younger pedestrian walking speeds. *Road Engineering Journal*.

TREUILLE, A., COOPER, S., AND POPOVIĆ, Z. 2006. Continuum crowds. *ACM Trans. Graph. 25*, 3 (July), 1160–1168.

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.

VANEGAS, C. A., ALIAGA, D. G., BENES, B., AND WADDELL, P. 2009. Interactive design of urban spaces using geometrical and behavioral modeling. *ACM Trans. Graph. 28*, 5, 111:1–111:10.

VANEGAS, C. A., GARCIA-DORADO, I., ALIAGA, D. G., BENES, B., AND WADDELL, P. 2012. Inverse design of urban procedural models. *ACM Trans. Graph. 31*, 6 (Nov.), 168:1–168:11.

VANEGAS, C. A., KELLY, T., WEBER, B., HALATSCH, J., ALIAGA, D. G., AND MÜLLER, P. 2012. Procedural generation of parcels in urban modeling. *Computer Graphics Forum 31*, 2pt3 (May), 681–690.

WEBER, B., MLLER, P., WONKA, P., AND GROSS, M. 2009. Interactive geometric simulation of 4d cities. *Computer Graphics Forum 28*, 2, 481–492.

WILKIE, D., SEWALL, J., AND LIN, M. 2013. Flow reconstruction for data-driven traffic animation. *ACM Trans. Graph. 32*, 4 (July), 89:1–89:10.

XU, K., ZHANG, H., COHEN-OR, D., AND CHEN, B. 2012. Fit and diverse: Set evolution for inspiring 3d shape galleries. *ACM Trans. Graph. 31*, 4 (July), 57:1–57:10.

YANG, Y.-L., WANG, J., VOUGA, E., AND WONKA, P. 2013. Urban pattern: Layout design by hierarchical domain splitting. *ACM Trans. Graph. 32*, 6 (Nov.), 181:1–181:12.

YU, Q., AND TERZOPOULOS, D. 2007. A decision network framework for the behavioral animation of virtual humans. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '07, 119–128.

YU, L.-F., YEUNG, S.-K., TANG, C.-K., TERZOPOULOS, D., CHAN, T. F., AND OSHER, S. J. 2011. Make it home: Automatic optimization of furniture arrangement. *ACM Trans. Graph. 30*, 4 (July), 86:1–86:12.

YU, L.-F., YEUNG, S.-K., TERZOPOULOS, D., AND CHAN, T. F. 2012. Dressup!: Outfit synthesis through automatic optimization. *ACM Trans. Graph. 31*, 6 (Nov.), 134:1–134:14.