Data-Driven Agent-Based Simulation for Pedestrian Capacity Analysis

Sing Kuang Tan[1(✉)], Nan Hu[2], and Wentong Cai[1]

[1] School of Computer Science and Engineering, Nanyang Technological University, Singapore, Singapore
{singkuang,aswtcai}@ntu.edu.sg
[2] Institution of High Performance Computing, Agency for Science Technology and Research, Singapore, Singapore
hun@ihpc.a-star.edu.sg

Abstract. In this paper, an agent-based data-driven model that focuses on path planning layer of origin/destination popularities and route choice is developed. This model improves on the existing mathematical modeling and pattern recognition approaches. The paths and origins/destinations are extracted from a video. The parameters are calibrated from density map generated from the video. We carried out validation on the path probabilities and densities, and showed that our model generates better results than the previous approaches. To demonstrate the usefulness of the approach, we also carried out a case study on capacity analysis of a building layout based on video data.

1 Introduction

Capacity analysis is to measure of the amount of pedestrian traffic a building layout can handle. To apply crowd simulation models in real applications, we can vary the inflow of people into a building layout to determine the capacity of the amount of pedestrian traffic the layout can handle by measuring the pedestrians' speeds and densities. It can be used to detect congested regions, and underutilized regions in a building layout. And these can be further used to evaluate different policies for crowd management and optimization (e.g., it can be used for event planning when a large crowd is expected). In summary, capacity analysis is useful to measure the effectiveness of a layout and plans for upgrading layout or managing the crowd.

Existing works on capacity analysis using agent-based simulation specify the pedestrians' movement rules in a layout manually [16,17]. Then the density distribution of the pedestrians is analyzed to determine the bottlenecks in the layout. Molyneaux et al. [8] proposed pedestrian management strategies such as the use of access gate and flow separation. Fundamental diagram [13] can be used to assess the capacity of a building layout and crowd management policy. Metrics [10] such as speed, travel time and level-of-service are used. Current works *use manually defined routes to do simulation for capacity analysis*. They

Y. Shi et al. (Eds.): ICCS 2018, LNCS 10861, pp. 103–116, 2018.
https://doi.org/10.1007/978-3-319-93701-4_8

only analyze speeds and densities in fundamental diagram, *ignoring the origin/destination (OD) popularities*. We developed more sophisticated metric to *analyze the histogram of density distributions* (see Sect. 4.3) instead of instantaneous density [5] or average density [16,17] in previous works. By deriving interpersonal distances from densities, we can understand the safety and comfort of the pedestrians better.

Using agent-based modeling and simulation for capacity planning has many advantages over previous methods of mathematical analysis using statistical route choices [9,12]. It can model the effect of changes in the environment, e.g., adding a new obstacle that lies in the walking paths of the pedestrians; and the detailed crowd behaviors such as group behaviors and inter-personal collision avoidance which the mathematical modeling approach cannot handle. As collision avoidance behavior is generally well studied [4,7] and data-driven path planning presents a more challenging research issue to form realistic crowd dynamics, we focus our study here on learning the route choice preference and the preference of selecting the origins (O) and destinations (D) in the layout. We formulate the OD popularities, and route choice model between a given OD pair in this work. Parameters of our model are calibrated through differential evolution genetic algorithm (GA) using a crowd density map extracted from KLT tracks [11]. Then from the learned parameters, capacity analysis is carried out on the layout.

The following components are generally required in agent-based simulation for capacity planning: identification of OD and routes, route choice model, and determination of OD popularities. With these components, pedestrian simulation can then be performed to get the pedestrian tracks. Capacity analysis metrics are then applied to the tracks to measure the amount of pedestrian traffic a building layout can handle.

The paper is organized as follows: Sect. 2 describes the related works. Section 3 describes our data-driven framework (OD and route identification, route choice model, pedestrian simulation and lastly parameters calibration). Section 4 presents a case study. Section 5 concludes this paper.

2 Related Works

Many crowd models have been proposed and developed over the years. For the high level behaviors of pedestrians, the choice of origin and destination using OD matrix [1] and the preference of different routes due to their differences in lengths and differential turns using statistical route choice [9] can be used. There is also a vector field model that maps each pedestrian position to the velocity vector based on the position of the pedestrian in the building layout [21]. A model of the adaption of each pedestrian speed and direction according to the distances and angles to nearby obstacle and destination [20] is created through genetic programming. For the low level behaviors of pedestrians, there are social force model [7] and RVO2 model [4]. Existing work learns route choice from density maps using mathematical modeling and optimization [12], which cannot

model the dynamic behavior of the pedestrians such as the obstacle collision avoidance behavior when an obstacle is added to the simulation. Unlike the existing mathematical route choice models that model the average statistical behavior of pedestrians over time, our model can simulate the instantaneous behaviors of agents with more precise positions than a discrete position layout used in mathematical modeling.

Recently there is a trend towards data-driven based approach to model crowd and calibrate model parameters. For calibrating interpersonal collision avoidance model parameters from videos, there is an anomaly detection approach [2]. An approach that extracts example behaviors from videos and use these examples to avoid collisions in agent-based pedestrian simulation is introduced in [19]. Interpersonal collision avoidance parameters can also be calibrated through laboratory experiments using deterministic approach [18] or non-deterministic approach [6]. Entry and exit regions transition probabilities can be learned either from the density maps [14] or from the KLT tracks [15]. Current works on data-driven modeling mostly focus on low-level pedestrian behavior models or do pattern recognitions on video or trajectories data. Instead of extracting patterns from data, we learn navigation behaviors of pedestrians that can be applied in an agent-based pedestrian simulation. This simulation can later be used to study different scenarios.

Crowd model parameters calibrations are often non-convex and require heuristic-based optimization algorithm such as genetic algorithm to search for good parameter values. Differential evolution genetic algorithm has shown to outperform many other variants of genetic algorithm on a wide set of problems [3]. In this paper, we followed similar approach as described in [22] to use differential evolution genetic algorithm and density-based calibration.

3 Data-Driven Framework

In this section, we will discuss about the framework of our data-driven agent-based pedestrian simulation model.

3.1 Overview of the Framework

The overview of our framework is shown in Fig. 1. A crowd simulation model is built based on empirical data extracted from videos, in particular, to capture the high-level motion of path planning through OD popularities and route choice modeling. The model is used to create agent-based simulation which is in turn used for capacity analysis of a given layout. It is conducted based on the calibrated simulation model. We will describe these in detail in the subsequent sub-sections.

To model the path planning behaviors of crowds, OD popularities and a route choice model for a given OD need to be determined. In this work, we focus on distilling OD popularities and calibrate route choice model parameters using video data.

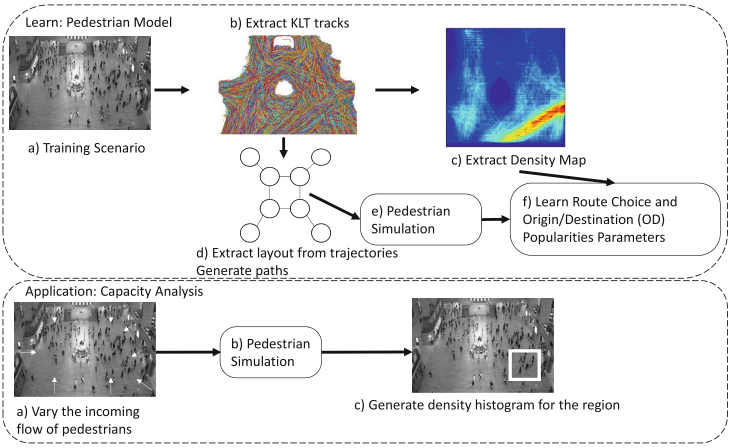


Fig. 1. The workflow of our framework from learning model to capacity analysis

3.2 OD and Path Identification

To get a full picture of the pedestrians in a building layout, the camera is preferably looking downward between 135 to 180° angle to the plane normal of the ground to minimize perspective distortion. The video can be in monochrome with a resolution high enough to get a few corner points on each pedestrian for tracking.

For a given video dataset, first image transformation is applied to remove perspective distortion of the camera. It is done by manually labeling some points in the ground plane in the video frame with the actual positions in the actual layout. The perspective transformation matrix is determined from the actual positions and pixel coordinates of the frame. Then an inverse perspective transform is applied on the video frame. The image transformation is also applied to the list of KLT tracks ρ_{KLT} (each track consists of a sequence of points (q_x, q_y), each of which is represented by $(\text{track\_id}, q_x, q_y, \text{time})$). Finally, we accumulate all the points in the KLT trajectories on a density map (grid size W by H) of the whole layout covered by the video. The density value at grid location (i, j) or distribution $\Pr(\mathrm{M}(i, j))$ is determined by:

$$\Pr(\mathrm{M}(i,j)) = \frac{1}{T} r^{\text{mask}}(i,j) \sum_{u=-h_{\text{size}}}^{h_{\text{size}}} \sum_{v=-h_{\text{size}}}^{h_{\text{size}}} \sum_{n} r_n(i+u, j+v) h(u,v) \tag{1}$$

$$T = \sum_{i,j} r^{\text{mask}}(i,j) \sum_{u=-h_{\text{size}}}^{h_{\text{size}}} \sum_{v=-h_{\text{size}}}^{h_{\text{size}}} \sum_{n} r_n(i+u, j+v) h(u,v) \tag{2}$$

$$r^{\text{mask}}(i,j) = \mathbf{1}_{\sum_n r_n(i,j)>0} \tag{3}$$

$$i = \{1, 2, \ldots, W\} \text{ and } j = \{1, 2, \ldots, H\} \tag{4}$$

where $r_n(i,j) = 1$ if track n passes through grid position (i,j). $\mathbf{1}$ is an indicator function which is 1 when the condition is true, else it is 0. $h(u,v)$ represents the smoothing filter of size h_{size}. Note that each track contributes one density count to a grid point in the density map and the points on each track are interpolated so that it is continuous. The density value is then normalized by the total density values so that it becomes a probability distribution. The grid points of the density map that are zeros form the mask map (r^{mask}) and these grid points are not used for calibrating the model parameters. These mask regions represent the walls and other barriers in the layout that the pedestrians cannot move into. The smoothing function $h(u,v)$ can be a Gaussian or uniform function.

The high density regions of the transformed ρ_{KLT} of a building layout are extracted by clustering all the (q_x, q_y) positions from the tracks using a Gaussian Mixture Modeling (GMM) algorithm as waypoints. The entrances of the layout (OD) can also be extracted by clustering. The number of clusters is selected using the elbow method by increasing the number of clusters until there is no significant increase in the maximum likelihood value of the clustering result. The W by H grid points of the layout is broken down into voronoi regions where each grid point is labeled to the nearest waypoint center and each mask region remains unlabeled without assigning to any waypoint. Two waypoint voronoi regions are adjacent if the pedestrian can walk from the first waypoint to the second waypoint without transversing other waypoints. We link the adjacent waypoints (voronoi) to form a topology map of the layout. For all pairs of OD, all possible paths (paths without repeating nodes) are generated between the OD.

3.3 Path Selection Model

Distance and turn distance are the commonly used path descriptors as the choice of path by the pedestrian is highly dependent on these two descriptors. These two descriptors are revised from [12]. The path descriptors of each path (p), namely the distance and turn distance, are computed using the formulas as follows:

$$\text{desc}_{\text{dist}}(\text{p}) = \frac{\sum_{i=1}^{N-1} \sqrt{(q_x^{(i+1)} - q_x^{(i)})^2 + (q_y^{(i+1)} - q_y^{(i)})^2}}{\sqrt{(q_x^{(N)} - q_x^{(1)})^2 + (q_y^{(N)} - q_y^{(1)})^2}} - 1 \tag{5}$$

$$\text{desc}_{\text{turn\_dist}}(\text{p}) = \frac{1}{\Pi} \sum_{i=1}^{N-2} \min(|\text{angle}_{i+2} - \text{angle}_{i+1}|, 2\pi - |\text{angle}_{i+2} - \text{angle}_{i+1}|) \tag{6}$$

$$\text{angle}_i = tan^{-1}\left(\frac{q_y^{(i)} - q_y^{(i-1)}}{q_x^{(i)} - q_x^{(i-1)}}\right) \tag{7}$$

where N is the number of waypoints for path p, $(q_x^{(i)}, q_y^{(i)})$ is the centroid position of the i-th waypoint of p and angle_i is the direction (in radians) between the waypoints $i-1$ and i. O and D centroids will be $(q_x^{(1)}, q_y^{(1)})$ and $(q_x^{(N)}, q_y^{(N)})$

respectively. The path descriptors distance and turn distance are normalized by the straight line distance between the OD and π respectively so that the descriptors are invariant to the scale size of the layout. We added these normalization techniques to the path descriptors introduced in [12] to improve learning performance.

The probability of taking p given o and d is then formulated as $\Pr(p|o,d)$ function as below,

$$\Pr(p|o,d) = \frac{\text{Pref}(p)}{\sum_{p' \text{ between } o \text{ and } d} \text{Pref}(p')} \tag{8}$$

$$\text{Pref}(p) = e^{\alpha \times \text{desc}_{\text{dist}}(p) + \beta \times \text{desc}_{\text{turn\_dist}}(p)}. \tag{9}$$

$\Pr(o,d)$ is the probability of selecting a pair of OD. $\text{Pref}(p)$ is preference of taking a particular path and it has a value between zero to positive infinity. In the expression $\Pr(p|o,d)$, the preference is normalized to a probability value between zero and one. The parameters α and β are to be learned empirically through the GA described later. The frequency of selecting p (number of times p is selected per second), $f(p)$ is therefore

$$f(p) = \sum_{o \in \text{O}, d \in \text{D}} \Pr(p|o,d) f(o,d) \tag{10}$$

where $f(o,d)$ is the frequency of selecting a pair of OD, which will be also learned through GA.

3.4 Parametrized Pedestrian Simulation

For each origin o, the simulation algorithm will generate a number of agents to be added to o using a Poisson distribution

$$n \sim \frac{e^{-k} k^n}{n!} \tag{11}$$

where $k = f(o) = \sum_{d \in \text{D}} f(o,d)$ and $f(o,d)$ (i.e., OD popularity) is a value in the simulation parameters. The destination of the agent a_i will be set according to

$$\Pr(d|\text{O}(a_i)) = \frac{f(\text{O}(a_i), d)}{\sum_{d' \in \text{D}} f(\text{O}(a_i), d')} \tag{12}$$

where $\text{O}(a_i)$ is the origin of agent a_i. These parameters are evolved by the GA to find a good set of values. The parameters will be described in more detail in the next section. For a layout of m entrances, there are $\frac{m(m-1)}{2}$ pairs (the permutation of arbitrary two out of m entrances) of OD. We assume that the o and d for each agent cannot be the same, and for a given (o,d) pair, agents have the same probability moving from o to d and from d to o. This assumption is made so as to keep the set of the OD popularities parameters smaller and

manageable. It also leads to better learning by preventing the creation of an overparameterized model.

For each origin o, new agents are added to the simulation at a fixed (i.e., every 5 s) interval according to Eq. (11). The destination (d) and path (p) of each agent is selected according to Eq. (12) and Eq. (8) respectively. They are assigned with the list of waypoints of $p \in P$ from o to d. The particular position (a waypoint is represented as a 2D Gaussian distribution learned from GMM) is selected randomly within the Gaussian distribution range of the waypoint,

$$(q_x, q_y) \sim \sqrt{\det(2\pi\mathbf{\Sigma_j})} e^{-\frac{1}{2}(\mathbf{q}-\mu_\mathbf{j})^T \mathbf{\Sigma_j}^{-1}(\mathbf{q}-\mu_\mathbf{j})} \tag{13}$$

where $\mu_\mathbf{j}$ and $\mathbf{\Sigma_j}$ are derived from GMM clustering, and $\mathbf{q}$ is the vector form of (q_x, q_y). Each agent is then following $p \in P$ from o through a list of waypoints to d. Agents avoid each other using a collision avoidance mechanism while moving between two consecutive waypoints. In this study, we apply the Reciprocal Velocity Obstacle (RVO2) method [4] for collision avoidance. RVO2 collision avoidance algorithm basically finds the best velocity vector for each agent to avoid collision. Once an agent reaches d, it will be removed from the simulation. Agents' trajectories through simulation are then aggregated. The density map is then created from the agents' trajectories in the same way as from the ρ_{KLT}. The detail description of our agent-based simulation procedure is shown in Fig. 2.

3.5 Path Selection Parameter and OD Popularity Determination

Our goal is to develop an agent-based model that behaves similarly to the video by having the same density distribution. In this model, we focus on the path planning layer of behaviors, which needs to set the route choice and OD popularities. The route choice and OD popularities will be the parameters to be calibrated by our GA. (Differential evolution) GA is very suitable for this problem as the cost function is non-convex. GA will reduce the number of simulation runs needed to do global optimization and it is important as each simulation run is a time-consuming process. As the parameters space is bounded by a set of minimum and maximum ranges instead of discrete values, this also makes GA very suitable.

First a population of random parameters are generated. The parameters are ordered in this particular order, $<\{f(o,d)|o \in \mathrm{O}, d \in \mathrm{D}\}, \alpha, \beta>$ where (α, β) are the route choice parameters. Then the fitness value of every individual of the population is calculated by running simulations using the parameter values of the individual, and compare the simulated density map with the ground truth density map using the formula below:

$$\text{fitness}, \lambda = \sqrt{\sum_{i=1}^{W}\sum_{j=1}^{H}(\Pr(\mathrm{M}(i,j)|\rho_{simulate}) - \Pr(\mathrm{M}(i,j)|\rho_{KLT}))^2} \tag{14}$$

Our Pedestrian Simulation

Input:
$f(o)$: Frequency of selecting a particular o
$\Pr(d|o)$: The probability of selecting a d given o
$\Pr(p|o,d)$: The probability of selecting a path p of a pair of OD
Return: the list of tracks $\rho_{simulate}$
Agent Generation Procedure:
for Every small time interval (i.e. 5 seconds interval) **do**
for Every origin o in layout **do**
Generate n number of agents using a Poisson distribution, Eq.(11)
Set the origin of each generated agent to o
Set the position of each generated agents to o position
Put these generated agents into the simulation
end for
end for
Agent Navigation Procedure:
for Each active agent a_i with $\mathrm{id} = \mathrm{id}(a_i)$ and $o = \mathrm{O}(a_i)$ **do**
Select the destination $\mathrm{D}(a_i)$ for agent a_i using $\Pr(d|\mathrm{O}(a_i))$, Eq.(12)
Select a path for agent a_i using $\Pr(p|\mathrm{O}(a_i), \mathrm{D}(a_i))$
for For every waypoints w_j on the path **do**
Generate a position (q_x, q_y) on the waypoint using Eq.(13)
Move agent a_i to position (q_x, q_y)
Record the track of agent, $(\mathrm{id}(a_i), q_x, q_y, \mathrm{time})$ into $\rho_{simulate}$
if Agent a_i reached the destination $\mathrm{D}(a_i)$ **then**
Remove the agent a_i
end if
end for
end for

Fig. 2. Procedure of our pedestrian simulation

where $\Pr(\mathrm{M}(i,j))$ is the probability of finding an agent/a pedestrian on a grid point (i,j) of the density map, W and H are the width and height of the density map. Note that $\Pr(\mathrm{M})$ sums to one and greater than zero and the mask regions of the density map are not used for parameter calibration. We use a probability distribution for the density map because we do not have the density values from the KLT tracks, but the relative densities between the grid points. As usual the population parameter values are evolved using differential evolution mutation and crossover methods to generate new offsprings. The fitness of these offsprings are evaluated using simulations and the fitness formula above. The offsprings will replace their parents if their fitness values are smaller than their parents. After several generations, the population will converge to a good set of parameter values.

4 Case Study

In this section, we will describe our scenario, evaluate our framework and lastly carry out capacity analysis using our framework.

4.1 Scenario Description

An agent-based crowd simulation, performing the path planning of crowds through the proposed route choice and OD popularities model, is developed in Java for the Grand Station dataset [23]. This dataset consists of a 33 minutes and 20 s video containing 50010 frames with a framerate of 25 fps at the resolution 720×480. A set of about 40000 KLT tracks, ρ_{KLT}, is also provided with the dataset. The GA is implemented in Matlab and for each set of parameter values, a multiple instances of the crowd simulation are executed. The average result over 4 runs is used for fitness evaluation. In this case, there are 8 entrances and therefore we have 28 pairs of OD. And another two route choice parameters, so we have in total 30 parameters. We choose a population size of 30 for the GA (we have also experimented with a population size of 100 and it leads to similar fitness value). We set the size of the density map to be 100 by 100 grid points to make it more manageable.

4.2 Evaluation of the Proposed Framework

In this section, we will compare our model (Model) against three baseline models: uniform OD popularity and shortest path (UniMod), existing vector-field model (VecMod) [21], and existing pedestrian-obstacle-destination model (PodMod) [20]. The ground truth (GT) will be derived from the ρ_{KLT}.

Figure 3 shows the density maps generated from our model and other existing approaches. We applied a small 5 by 5 window average filter to the density map (i.e., $h(u, v) = 1$ and $h_{\mathrm{size}} = 2$, see Eq. (4)) to filter out the randomness. Our approach matches the ground truth density map better than other approaches by more than 10% (by comparing the fitness values in the figure). As VecMod learns the path of the pedestrian from the directions of the ρ_{KLT} instead of from the density map of the ρ_{KLT}, it cannot model the variations of movements across the open space as well as our route choice approach. PodMod learns a deterministic function of movement for each OD pair, it only allows the pedestrian to move along one path instead of probabilistically select one of the paths in our route choice approach.

OD popularities parameters are calibrated by GA and simulation. The popularities can be estimated from the density map because the density between a high popularity OD will be higher and likewise the density between a low popularity OD will be lower. As for the OD popularities, Fig. 4(a) shows the relative popularity of each pair OD and Fig. 4(b) shows the density map obtained from the training video without applying any smoothing function (i.e., $h(u, v) = 0$ and $h_{\mathrm{size}} = 0$, see Eq. (4)). The high popularities between the bottom and right entrances further confirm what is shown in the video.

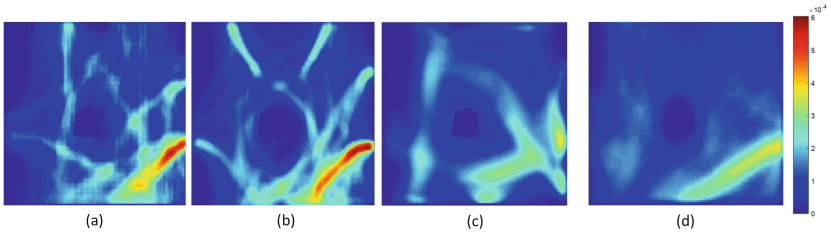


Fig. 3. Density maps generated by (a) VecMod [21], (b) PodMod [20], (c) our model and (d) GT. (Fitness, $\lambda =$ (a) 6.159×10^{-3} (b) 6.329×10^{-3} (c) $\mathbf{4.216 \times 10^{-3}}$)

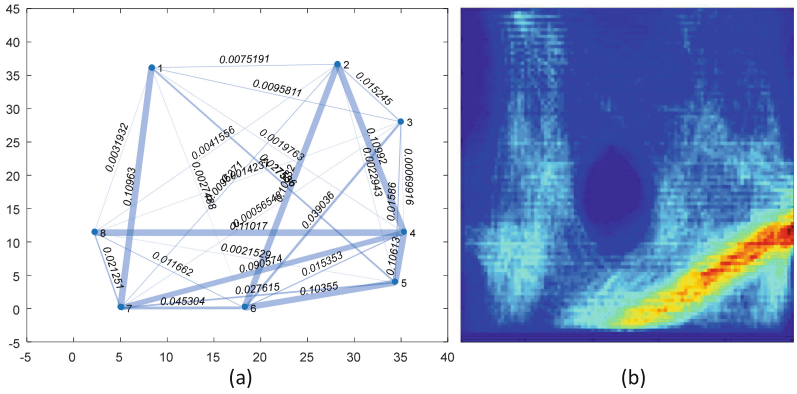


Fig. 4. (a) Relative popularities of learned OD popularities, (b) GT density map without applying any smoothing filter (see text for more details)

We compared the learned path probabilities with the path probabilities of the ρ_{KLT}. As the ρ_{KLT} are broken without OD information, we cannot directly map each track to a specific path. So we match each track to all paths with which the track matches partially, and evenly distribute the probabilities of the tracks to the matching list of paths. To specify it formally,

$$\Pr(p = \text{path}_i|\text{GT}) = \frac{1}{\alpha_i} \text{ if } \alpha_i > 0, \text{else } 0 \tag{15}$$

where α_i = # of tracks in ρ_{KLT} match sub-path of path_i and a KLT track matches sub-path of path_i if the track contains a 'substring' of the path_i's waypoints. The following distance functions are used for comparison:

$$\begin{aligned} \text{Total Variation Distance} &= \sum_i |\Pr(p = \text{path}_i|\text{Model}) - \Pr(p = \text{path}_i|\text{GT})| \\ \text{Histogram Intersection} &= \sum_i min(\Pr(p = \text{path}_i|\text{Model}), \Pr(p = \text{path}_i|\text{GT})). \end{aligned} \tag{16}$$

These two distance functions are commonly used for comparing between two probability distributions (it is the lower the better for variation distance; whereas it the higher the better for histogram intersection). UniMod is used as a baseline model as it is commonly assumed if we have no information of how often one pedestrian will choose a pair of OD over another pair.

Our model is better in terms of the two distance functions than the baseline UniMod. The distances (GT versus our model/GT versus UniMod) for total variation and histogram intersection are **1.9624**/1.9965 and **0.0188**/0.0017 respectively. The popularities across different pairs of OD are non-uniform as we observed that there are much more people walking from some of the entrances.

4.3 Capacity Analysis

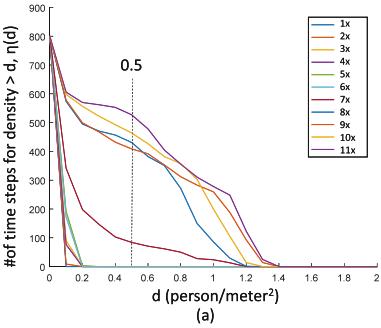
Following the work described in [10], we choose three metrics for capacity analysis,

$$\begin{aligned} \text{Density Distribution}, \eta(d) &= \sum_{t} \mathbf{1}_{\text{density}(t)>=d} \\ \text{Average Travel Speed}, \theta &= \frac{1}{M} \sum_{i=1}^{M} \text{Speed}(a_i) \\ \text{Travel Speed Index}, \vartheta &= \frac{\theta}{\theta_{\text{free\_flow}}} \end{aligned} \tag{17}$$

where density(t) is the density of the region at time t, Speed(a_i) is the speed of agent i, M is the number of agents in the region and $\theta_{\text{free\_flow}}$ is the average speed of the agents when the density is 0. $\eta(d)$ is the number of time steps where the density is greater than or equal to a specified amount d. $\eta(d)$ is selected because it has been used to determine the safety and comfort of the pedestrians [5]. θ is selected as it can tell us the time taken for a pedestrian to move through the region and give us the level of congestion. ϑ gives us the percentage of additional time that is needed to move through the crowded region compared to when the region has no crowd.

We varied the OD popularities by multiplying them using a fixed constant between 1 to 11. Figure 5(a) shows $\eta(d)$ (time step = 0.25 s). Figure 5(b) shows θ and ϑ. Figure 5(c) shows the region where the density and speeds are inspected for the different OD popularity values. This region is selected as it lies along the highest density path when the popularities are at normal values.

As the popularities get higher, the total number of people increases linearly, but the density increases non-linearly. The changes in the density (Fig. 5(a)) is non-linear and there is a tipping point of significant increase when the popularities increase from 7 to 8 times. The increase in density starts to slow down after 8 times. For instance, for $\eta(0.5)$, when the popularities are increased from 7 to 8 times, the frequency increases by more than 4 times. This makes intuitive sense as the density increases, the speeds of pedestrians decrease due to more collision avoidance and this in turn leads to larger increase in density. As the



Origin-destination (OD) Popularities	Avg speed (meter/s), θ	Avg speed/ free flow speed, ϑ	Level of service (LOS)
1x	1.0240	1.0186	A
2x	0.9820	0.9769	A
3x	0.9021	0.8974	A
4x	0.7334	0.7295	A
5x	0.6898	0.6862	A
6x	0.5298	0.5271	A
7x	0.4561	0.4537	D
8x	0.4355	0.4332	D
9x	0.4094	0.4073	D
10x	0.3758	0.3739	D
11x	0.3513	0.3495	D

(b)

(c)

Fig. 5. (a) Density and (b) speed changes due to increase in OD popularities. (c) Region under analysis

density further increases, jams occur at some parts of the layout and this reduces the rate of increment of the density at the region under study. The capacities at different regions are also affected by layout structure which determines where and how density is accumulated. This kind of dynamic behavior is difficult to model mathematically and the results are different for different layouts.

We can also see that as the popularities get higher, θ decreases, where the rate of decreases is higher between 3 to 7 times of normal popularities. This is due to the same observation as the density. However the decrease in speed is not as obvious as the increase in density. For the level of service (LOS) [5], it is 'A' (free circulation) when the increase of popularity is below 7 times, but it changes drastically to 'D' (restricted and reduced speed for most pedestrians) when the increase of popularity is above or equal 7 times. For ϑ, a value of 1 indicates that the average travel speed is at its optimal speed and is not affected by the density (due to small randomness in the simulation, ϑ can be slightly larger than 1 as in the 1st row of the table).

5 Conclusion

We have developed a data-driven agent-based framework that focuses on the path planning layer. And this framework can be used for capacity analysis. We have carried out experiments and analysis on the learned parameters and density map of our model, performed capacity analysis on hypothetical situation where the OD popularities were varied by a constant multiplier.

The model created can be used for analyzing different crowd management policies, sudden increase in crowd densities, and other novel scenarios. In the future, we will automate crowd management strategies through optimization of speeds of the pedestrians at different locations or re-routing the pedestrians, enforced by marshallers on the ground.

The assumption we make here is that as density increases uniformly, people's path planning is not affected much by the density increment, but still by

space syntax (layout). There is one imperfection in our model is that it does not model change in a pedestrian route due to very high density congestion. Congestion model is important as we continuously increase the number of agents in the simulation for capacity analysis, it will definitely lead to very serious congestion at some point. As our future work, we will add congestion model into the current route choice model to model the change of pedestrian behaviors during congestion to tackle this problem. We are also planning to use virtual reality experiments to collect data under controlled environment.

Acknowledgement. Singkuang Tan, Nan Hu, and Wentong Cai would like to acknowledge the support from the grant: IHPC-NTU Joint R&D Project on "Symbiotic Simulation and Video Analysis of Crowds".

References

1. Asakura, Y., Hato, E., Kashiwadani, M.: Origin-destination matrices estimation model using automatic vehicle identification data and its application to the Han-Shin expressway network. Transportation **27**(4), 419–438 (2000)
2. Charalambous, P., Karamouzas, I., Guy, S.J., Chrysanthou, Y.: A data-driven framework for visual crowd analysis. In: CGF, vol. 33, pp. 41–50. Wiley Online Library (2014)
3. Das, S., Suganthan, P.N.: Differential evolution: a survey of the state-of-the-art. TEVC **15**(1), 4–31 (2011)
4. Fiorini, P., Shiller, Z.: Motion planning in dynamic environments using velocity obstacles. IJRR **17**(7), 760–772 (1998)
5. Fruin, J.J.: Pedestrian planning and design. Technical report (1971)
6. Guy, S.J., Van Den Berg, J., Liu, W., Lau, R., Lin, M.C., Manocha, D.: A statistical similarity measure for aggregate crowd dynamics. TOG **31**(6), 190 (2012)
7. Helbing, D., Molnár, P.: Social force model for pedestrian dynamics. Phys. Rev. E **51**, 4282–4286 (1995)
8. Molyneaux, N., Scarinci, R., Bierlaire, M.: Pedestrian management strategies for improving flow dynamics in transportation hubs. In: STRC (2017)
9. Prato, C.G.: Route choice modeling: past, present and future research directions. J. Choice Model. **2**(1), 65–100 (2009). https://doi.org/10.1016/S1755-5345(13)70005-8. http://www.sciencedirect.com/science/article/pii/S1755534513700058
10. Rao, A.M., Rao, K.R.: Measuring urban traffic congestion-a review. IJTTE **2**(4) (2012)
11. Shi, J., Tomasi, C.: Good features to track. In: CVPR, pp. 593–600 (1994). https://doi.org/10.1109/CVPR.1994.323794
12. Tan, S.K.: Visual detection and crowd density modeling of pedestrians. Ph.D. thesis, SCSE, NTU (2017). http://hdl.handle.net/10356/72746
13. Vanumu, L.D., Rao, K.R., Tiwari, G.: Fundamental diagrams of pedestrian flow characteristics: a review. ETRR **9**(4), 49 (2017)
14. Wang, H., Ondřej, J., O'Sullivan, C.: Trending paths: a new semantic-level metric for comparing simulated and real crowd data. TVCG **23**(5), 1454–1464 (2017)
15. Wang, H., O'Sullivan, C.: Globally continuous and non-Markovian crowd activity analysis from videos. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9909, pp. 527–544. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46454-1_32

16. Wang, H., Yu, L., Qin, S.: Simulation and optimization of passenger flow line in Lanzhou West Railway Station. In: Sierpiński, G. (ed.) TSTP 2017. Advances in Intelligent Systems and Computing, vol. 631, pp. 61–73. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-62316-0_5
17. Wang, R., Zhang, Y., Yue, H.: Developing a new design method avoiding latent congestion danger in urban rail transit station. Transp. Res. Procedia **25**, 4083–4099 (2017)
18. Wolinski, D., J Guy, S., Olivier, A.H., Lin, M., Manocha, D., Pettré, J.: Parameter estimation and comparative evaluation of crowd simulations. In: CGF, vol. 33, pp. 303–312. Wiley Online Library (2014)
19. Zhao, M., Turner, S.J., Cai, W.: A data-driven crowd simulation model based on clustering and classification. In: DS-RT, pp. 125–134. IEEE (2013)
20. Zhong, J., Cai, W., Lees, M., Luo, L.: Automatic model construction for the behavior of human crowds. Appl. Soft Comput. **56**, 368–378 (2017). https://doi.org/10.1016/j.asoc.2017.03.020
21. Zhong, J., Cai, W., Luo, L., Yin, H.: Learning behavior patterns from video: a data-driven framework for agent-based crowd modeling. In: AAMAS, pp. 801–809 (2015). http://dl.acm.org/citation.cfm?id=2773256
22. Zhong, J., Hu, N., Cai, W., Lees, M., Luo, L.: Density-based evolutionary framework for crowd model calibration. J. Comput. Sci. **6**, 11–22 (2015)
23. Zhou, B., Wang, X., Tang, X.: Understanding collective crowd behaviors: learning a mixture model of dynamic pedestrian-agents. In: CVPR, pp. 2871–2878. IEEE (2012)