A data-driven path planning model for crowd capacity analysis

Tan, Sing Kuang; Hu, Nan; Cai, Wentong

2019

Tan, S. K., Hu, Nan, & Cai, W. (2019). A data-driven path planning model for crowd capacity analysis. *Journal of Computational Science*, 34, 66-79. doi:10.1016/j.jocs.2019.05.003

<https://hdl.handle.net/10356/143350>

<https://doi.org/10.1016/j.jocs.2019.05.003>

© 2019 Elsevier B.V. All rights reserved. This paper was published in *Journal of Computational Science* and is made available with permission of Elsevier B.V.

Downloaded on 26 Apr 2023 22:54:46 SGT

A Data-driven Path Planning Model for Crowd Capacity Analysis

Sing Kuang Tan

Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore

Nan Hu

Alibaba Cloud, Singapore

Wentong Cai

School of Computer Science and Engineering, Nanyang Technological University, Singapore

Abstract

In this paper, an agent-based crowd simulation model that focuses on path planning layer of 1) origin/destination popularities and 2) route choice is developed. This path planning model improves on the existing mathematical modeling and pattern recognition approaches by utilizing different sources of data to drive and validate it: video data was used for the open space scenarios and virtual reality experiments were applied for constrained space scenarios. For open space scenarios with video coverage, the density map of the video is extracted to calibrate the origin/destination popularities and the route probabilities among them. Factors related to space syntax, such as the traveling distance and turning angle, are proven effective features of the path planning model in this scenario. For constrained space scenarios, where the coverage of videos is usually limited, virtual reality experiments can be applied to learn the route choice model parameters at a fine granularity, particularly considering the crowdedness of the surroundings besides the space syntax factors. The navigation behaviors of players under different configurations in the virtual reality experiments were retrieved to train the route choice models using Support Vector Machine (SVM) model. The trained route choice model then simulates the

crowd motion more realistically under different densities. We demonstrate the usefulness of the data-driven path planning model for crowd capacity analysis of a building layout.

Keywords: Data-Driven Modeling, Crowd Counting and Tracking, Route Choice, Pedestrian Simulation, Virtual Reality, Crowd Avoidance Behavior, Support Vector Machine

1. Introduction

Capacity analysis measures the amount of pedestrian traffic a building layout can handle. Crowd simulation models are usually applied for capacity analysis. With various inflow of people into a building layout as the model inputs, crowd
5 motion is simulated to determine the capacity of pedestrian traffic at different locations by measuring the pedestrians' speeds and densities. Such simulation studies can be used to detect congested and underutilized regions in a building layout as applications for instance. Findings from such studies can be further used to evaluate different policies for crowd management and optimization (e.g.,
10 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 modeling and simulation specify the pedestrians' movement rules in a layout manually [1, 2], then the
15 density distribution of the pedestrians is analyzed to determine the bottlenecks in the layout. Molyneaux et al. [3] proposed pedestrian management strategies such as the use of access gate and flow separation. Fundamental diagram [4] can be used to assess the capacity of a building layout and crowd management policy. Metrics [5] such as speed, travel time and level-of-service are used. Cur-
20 rent works *use manually defined routes to do simulation for capacity analysis*. They only analyze speeds and densities in fundamental diagram, *ignoring the origin/destination (OD) popularities*. We develop more sophisticated metrics to *analyze the histogram of density distributions* (see section 4.3) instead of instan-

taneous density [6] or average density [1, 2] as reported in previous works. By
25 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 [7, 8]. It can model the effect of changes in the environment,
30 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 [9, 10] and data-driven
path planning presents a more challenging research issue to form realistic crowd
35 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 to properly model the path planning behaviors of crowd in this work.

We test the proposed path planning model in both open space and con-
40 strained space scenarios. In the open space scenario, we study a train station
with an open space and only one obstacle in the center (the information center).
Video footage is available in such a scenario¹, capturing the space of study. Pa-
rameters of our path planning model focusing on the OD popularity and route
choices are calibrated through a differential evolutionary genetic algorithm² us-
45 ing a crowd density map extracted from KLT tracks [11] of the video. From the
calibrated crowd simulation model with the learned parameters, capacity anal-
ysis is carried out on the layout. In the later part of this paper (i.e., constrained
space scenario), a pedestrian route choice model with crowd avoidance behav-
ior is developed. This model extends our data-driven modeling approach [12]

¹<http://www.ee.cuhk.edu.hk/~xgwang/grandcentral.html>

²Differential evolution and genetic algorithm are two separate algorithms. But differential evolution can be used in the mutation step of the genetic algorithm. We combined the two algorithms so that the final algorithm can have the advantages of both algorithms. We called it differential evolutionary genetic algorithm, DEGA.

that ignores the size of crowd density in a building layout. In the constrained space scenario, we explore a smaller layout of a restaurant, where the commonly adopted paths are fixed, and we study the effect of crowdedness on the route choice more precisely. Route choice parameters including both space syntax and crowdedness are calibrated using Support Vector Machine (SVM) from the experiment data collected from carefully designed virtual reality (VR) experiments. In common for both scenarios, space syntax regarding the layout of space will be considered in the path planning model. However, there are also differences for these scenarios: in the train station scenario, pedestrians can see their destinations directly and move towards them without much detour. But in the restaurant layout, pedestrians have to make decision at every junction due to the blockage of sight in the constrained space. Through the use of VR experiments with a virtual layout and strategically placed virtual crowds at different junctions on alternative paths, we can study and quantify the effect of crowdedness on their route choices.

The paper is organized as follows: Section 2 describes the related works. Section 3 describes our data-driven path planning model for crowd modeling and simulation (OD and route identification, route choice model, pedestrian simulation, and lastly parameter calibration). Section 4 presents a case study of crowd capacity analysis in the open space scenario, and Section 5 describes another case study in the constrained space scenario, with crowdedness considered in the route choice model. Section 6 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 [13] and the preference of different routes due to their differences in lengths and differential turns using statistical route choice [7] 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 [14].

A model that adapts each pedestrian speed and direction according to the distances and angles to nearby obstacle and destination [15] is created through
80 genetic programming. For the low level behaviors of pedestrians, there are social force model [9] and RVO2 model [10]. Existing work learns route choice from density maps using mathematical modeling and optimization [8], which cannot *model the dynamic behavior* of the pedestrians such as the obstacle col-
85 lision 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. Existing OD and route choice model assumes
90 that the crowd density in the layout does not affect the pedestrian choice, but this is not the case for high density scenario.

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 [16].
95 An approach that extracts example behaviors from videos and use these examples to avoid collisions in agent-based pedestrian simulation is introduced in [17]. Interpersonal collision avoidance parameters can also be calibrated through laboratory experiments using deterministic approach [18] or non-deterministic approach [19]. Entry and exit regions transition probabilities can be learned
100 from either the density maps [20, 21] or the KLT tracks [22]. 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. The learned behaviors can
105 later be used to study different scenarios. In our crowd avoidance route choice model, instead of learning through videos, we collect data through VR experiments which we can control the variables in the experiment unlike the case of using videos.

Crowd model parameters calibrations are often non-convex and require heuristic-

Figure 1: The workflow from learning model to capacity analysis

110 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 [23, 24]. In this paper, we follow similar approach as described in [25] to use differential evolution genetic algorithm and density-based calibration. We also use VR experiments [26] to validate our model. VR experiment has the advantage that we can directly observe the effect of one variable over another directly.

3. Data-Driven Path Planning Model

In this section, we will discuss about the proposed data-driven path planning model for agent-based crowd modeling and simulation.

120 3.1. Model Overview

The overview of the proposed model is shown in Figure 1. A crowd simulation model is built and calibrated based on empirical data extracted from videos or VR experiments, in particular, to capture the high-level motion of

path planning through OD popularities and route choice modeling. The crowd
125 simulation model is in turn used for capacity analysis of a given layout. 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 the route choice model parameters using
130 either video data or VR experiments.

3.2. OD Popularity 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 degrees angle to the plane
normal of the ground to minimize perspective distortion. The video can be in
135 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 time-
stamped points. 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 $\Pr(M(i, j))$ is determined

by:

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

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

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

$$i = \{1, 2, \dots, W\} \text{ and } j = \{1, 2, \dots, H\} \quad (4)$$

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
140 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
145 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/exits of the
150 layout (i.e., ODs) 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
155 region remains unlabeled without assigning to any waypoint. Two waypoints are adjacent if the pedestrian can walk from the first waypoint to the second waypoint without transversing other voronoi regions. We link the adjacent waypoints to form a topology map of the layout. For all OD pairs, all possible paths (paths without repeating nodes) are generated between each OD pair. To

160 this end, the popular OD and frequently used paths in a given scenario with
video footage can be identified.

3.3. Route Choice

Traveling distance and turning distance are the commonly used path descriptors to quantify preferences over different routes. These two descriptors
165 have been reviewed in [8]. In addition, the angle of path from the destination is also an important path descriptor. Based on the theory of space syntax, people tend to move with minimal detour effort towards their destinations. Besides factors related to space syntax of a layout, dynamic factors observed in the scenario such as the perceived crowdedness along different paths can also be
170 important to influence route choices. However, so far as we are aware, it is not well studied quantitatively in the context of crowd's path planning yet. In this paper, we consider descriptors related to both space syntax and the dynamic perceived factor such as the crowdedness, trying to model the route choices more realistically.

The path descriptors of each path (p), namely the distance, turn distance, angle of path from destination, and crowdedness (in terms of the crowd density), are computed using the formulas as follows:

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

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

$$\text{desc}_{\text{angle_from_dest}}(p) = \frac{1}{\Pi} \min(|\text{angle}_{2,1} - \text{angle}_{N,1}|, 2\pi - |\text{angle}_{2,1} - \text{angle}_{N,1}|) \quad (7)$$

$$\text{angle}_{i,j} = \tan^{-1} \left(\frac{q_y^{(i)} - q_y^{(j)}}{q_x^{(i)} - q_x^{(j)}} \right) \quad (8)$$

$$\text{desc}_{\text{density}}(p) = \frac{\# \text{ of people in the crowd}}{\text{Area of crowd}} \quad (9)$$

175 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 π is the mathematical Pi 3.14159. The path descriptors
distance and turn distance are normalized by the straight line distance between
180 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 [8] to improve learning performance.

The probability of taking p given o and d is then formulated as $\text{Pr}(p|o, d)$
function as follows:

$$\text{Pr}(p|o, d) = \frac{\text{Pref}(p)}{\sum_{p' \text{ between } o \text{ and } d} \text{Pref}(p')} \quad (10)$$

$$\text{Pref}(p) = e^{\alpha \times \text{desc}_{\text{dist}}(p) + \beta \times \text{desc}_{\text{turn_dist}}(p) + \gamma \times \text{desc}_{\text{angle_from_dest}}(p) + \delta \times \text{desc}_{\text{density}}(p)}. \quad (11)$$

$\text{Pref}(p)$ is preference of taking a particular path and it has a value between zero
to positive infinity. In the expression $\text{Pr}(p|o, d)$, the preference is normalized to a
probability value between zero and one. The parameters are learned empirically
as described later in the case studies. The frequency of selecting p (number of
times p is selected per second), $f(p)$ is therefore defined as follows:

$$f(p) = \sum_{o \in O, d \in D} \text{Pr}(p|o, d) f(o, d) \quad (12)$$

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

185 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:

$$n \sim \frac{e^{-k} k^n}{n!} \quad (13)$$

where $k = f(o) = \sum_{d \in \mathcal{D}} f(o, d)$ 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

$$\Pr(d|O(a_i)) = \frac{f(O(a_i), d)}{\sum_{d' \in \mathcal{D}} f(O(a_i), d')} \quad (14)$$

where $O(a_i)$ is the origin of agent a_i . These parameters are evolved by the DEGA 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
190 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 seconds) interval according to Eq.13. The destination (d) and path (p) of each agent is selected according to Eq.14 and Eq.10 respectively. They are assigned with the list of waypoints of $p \in P$ from o to d . The particular position
³ is selected randomly within the Gaussian distribution range of the waypoint:

$$(q_x, q_y) \sim \sqrt{\det(2\pi \Sigma_j)} e^{-\frac{1}{2}(\mathbf{q} - \mu_j)^T \Sigma_j^{-1} (\mathbf{q} - \mu_j)} \quad (15)$$

195 where μ_j and Σ_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 [10] for collision avoidance. RVO2 collision
200 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

³A waypoint is represented as a 2D Gaussian distribution learned from GMM.

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 Figure 2.

205 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
 210 be calibrated by our DEGA, which is very suitable for this problem as the cost function is non-convex. DEGA 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 DEGA very
 215 suitable. We can also use SVM to classify best path from other paths in route choice if we have the data of the routes taken by each pedestrian.

First a population of random parameters are generated. The parameters are ordered in this particular order, $\langle \{f(o, d) | o \in O, d \in D\}, \alpha, \beta, \gamma, \delta \rangle$ where $(\alpha, \beta, \gamma, \delta)$ 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:

$$\text{fitness}, \lambda = \sqrt{\sum_{i=1}^W \sum_{j=1}^H (\Pr(M(i, j) | \rho_{simulate}) - \Pr(M(i, j) | \rho_{KLT}))^2} \quad (16)$$

where $\Pr(M(i, j))$ 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 $\Pr(M)$ sums to one and greater than zero and the mask regions
 220 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

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.13
    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  $\text{id} = \text{id}(a_i)$  and  $o = O(a_i)$  do
  Select the destination  $D(a_i)$  for agent  $a_i$  using  $\Pr(d|O(a_i))$ , Eq.14
  Select a path for agent  $a_i$  using  $\Pr(p|O(a_i), D(a_i))$ 
  for For every waypoints  $w_j$  on the path do
    Generate a position  $(q_x, q_y)$  on the waypoint using Eq.15
    Move agent  $a_i$  to position  $(q_x, q_y)$ 
    Record the track of agent,  $(\text{id}(a_i), q_x, q_y, \text{time})$  into  $\rho_{simulate}$ 
    if Agent  $a_i$  reached the destination  $D(a_i)$  then
      Remove the agent  $a_i$ 
    end if
  end for
end for

```

Figure 2: Procedure of our pedestrian simulation

mutation and crossover methods to generate new offsprings. The fitness of these
225 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 1: Open Space Scenario

230 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 [27]. In this scenario, crowdedness is not
considered as a factor of route choice model, because the space is relatively open.
235 We will look into a constrained space layout in the next scenario and show that
our model can adapt to represent both scenarios. This dataset consists of a
33 minutes and 20 seconds video containing 50,010 frames with a framerate
of 25 fps at the resolution 720x480. A set of about 40,000 KLT tracks, ρ_{KLT} ,
is also provided with the dataset. The DEGA is implemented in Matlab and
240 for each set of parameter values, a multiple instances of the crowd simulation
are executed. The average result over 4 runs is used for the fitness evaluation.
In this case, there are 8 entrances and therefore we have 28 pairs of OD. In
this scenario, the angle of path from destination and crowd density, as the
pedestrian can simply weave through the crowd in this open area scenario. So,
245 with other two route choice parameters, we have in total 30 parameters. We
choose a population size of 30 for the DEGA (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.

In order to map our simulation to reality, the density distribution from the
250 pedestrian simulation will be matched against the density distribution of the
pedestrians in the video. We will select the parameters for the simulation that
matches the two density distributions best. It is assumed that if the density

distributions are similar, then the pedestrians in both simulation and reality will have the same probabilities for choosing their origin-destination and also have the same probabilities for choosing their routes out of all routes from their origins to destinations. With this computational pedestrian model of the reality, we can tweak the parameters of the model such as increasing the incoming rate of the pedestrians from the origins to study the congestion behavior. This allows us to study the congestion behavior without doing physical pedestrian experiment in the actual layout.

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) [14], and existing pedestrian-obstacle-destination model (PodMod) [15]. 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 $h_{\text{size}} = 2$, see Eq.4) to filter out the randomness. The main use of $h(u, v)$ is to smooth the density map created by discrete and finite pedestrian tracks which are random and slightly different across different runs of the same simulation. 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 DEGA 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, Figure 4(a) shows

Figure 3: Density maps generated by (a) VecMod [14], (b) PodMod [15], (c) our model and (d) GT. (Fitness, $\lambda =$ (a) 6.159×10^{-3} (b) 6.329×10^{-3} (c) **4.216×10^{-3}**)

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

the relative popularity of each OD pair and Figure 4(b) shows the density map obtained from the training video without applying any smoothing function (i.e., $h(u, v) = 0$ and $h_{\text{size}} = 0$, see Eq.4). The high popularities between the bottom and right entrances further confirm what is shown in the video.

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,

$$\Pr(p = \text{path}_i | \text{GT}) = \frac{1}{\alpha_i} \text{ if } \alpha_i > 0, \text{ else } 0 \quad (17)$$

where α_i = number 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:

$$\begin{aligned} \text{Total Variation Distance} &= \sum_i |\Pr(p = \text{path}_i | \text{Model}) - \Pr(p = \text{path}_i | \text{GT})| \\ \text{Histogram Intersection} &= \sum_i \min(\Pr(p = \text{path}_i | \text{Model}), \Pr(p = \text{path}_i | \text{GT})). \end{aligned} \quad (18)$$

290 These two distance functions are commonly used for comparing between two probability distributions (it is the lower the better for variation distance; whereas it is 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.

295 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.

300 4.3. Capacity Analysis

Following the work described in [5], we choose three metrics for capacity analysis,

$$\begin{aligned} \text{Density Distribution, } \eta(d) &= \sum_t \mathbf{1}_{\text{density}(t) \geq d} \\ \text{Average Travel Speed, } \theta &= \frac{1}{M} \sum_{i=1}^M \text{Speed}(a_i) \\ \text{Travel Speed Index, } \vartheta &= \frac{\theta}{\theta_{\text{free_flow}}} \end{aligned} \quad (19)$$

where $\text{density}(t)$ is the density of the region at time t , $\text{Speed}(a_i)$ 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

Figure 5: Route choice and path descriptors

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 [6]. 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 5a shows $\eta(d)$ (time step= 0.25 second). Figure 5b shows θ and ϑ . Figure 5c 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 (Figure 5a) 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 popularity increases, the speeds of pedestrians decrease due to more collision avoidance and this in turn leads to larger increase in density. As the popularity further increases, jams occur at some parts of the layout and this reduces the rate of increment of $\eta(d)$ 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) [6], 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. Case Study 2: Constrained Space Scenario

5.1. Scenario Description

In this scenario, we look into a restaurant layout where movements are constrained by the tables and chairs. Thus, this scenario has limited fixed number of routes/paths, and the crowdedness will cause the pedestrian to switch to another path.

Figure 6: Process to learn our Crowd Avoidance Model

5.2. Route Choice Model

345 In this section, we will develop the route choice model including the crowd-
 edness on a scenario created through VR experiments. Our process to learn
 this model is shown in Figure 6, first a VR experiment is carried out where the
 player is given an instruction to move from the starting point to the destination.
 When the player is at a specific junction, the player has two routes to choose
 350 from and a stationary crowd of random size will be placed on the shorter path
 near the junction. The junction, the routes, and the choice will be recorded
 down automatically by the VR software.

The route choice problem (see Figure 7a) is formulated as a binary clas-
 sification problem. Given the pedestrian at a given junction with two route
 choices, which route will the pedestrian select? The pedestrian will make the
 decision based on the distance, total amount of differential turns, angle of route
 from the destination(see Figure 7b) and the density of crowd on the route.
 Given the choice of shorter route and longer route, the shorter route is taken
 when $\text{Pref}(p_{\text{short}}) > \text{Pref}(p_{\text{long}})$, where p_{short} and p_{long} is the paths of the
 shorter and longer routes respectively. Likewise the longer route is taken when
 $\text{Pref}(p_{\text{short}}) < \text{Pref}(p_{\text{long}})$. This can be formulated as a linear SVM to learn the
 preference function $\text{Pref}(p_{\text{short}})$ and $\text{Pref}(p_{\text{long}})$ where the preference function

Figure 7: Route choice and path descriptors

is defined as follows:

$$\begin{aligned} \text{Pref}(p) = & \alpha \times \text{desc}_{\text{dist}}(p) + \beta \times \text{desc}_{\text{turn_dist}}(p) \\ & + \gamma \times \text{desc}_{\text{angle_from_dest}}(p) + \delta \times \text{desc}_{\text{density}}(p). \end{aligned} \quad (20)$$

Note that this is different from Eq.11. The exponential function is taken out as it is a monotonic function and it will not affect the comparison of the value of the preference function between the two routes. The linear combination of the route choice parameters and route choice descriptors can be further generalized to non-linear combination as in kernel SVM. There is no need to learn the topology map of the layout and waypoints as the paths in the scenario layout of the VR experiment are fixed and discrete. Since the route choice problem of this second case study is a classification problem, SVM is suitable for this classification task instead of Genetic algorithm, whereas Genetic algorithm is useful for parameter calibration task in the first case study.

Due to the use of a small layout (in order to study the effect of crowd density on route choice in detail), a deep learning classifier is not suitable as the variation of the datapoints is small. To study whether non-linearity in the data helps in the classification, we use kernel SVM. Using a sigmoid kernel SVM is equivalent to a 3-layer neural network [28], so we avoid experimenting with shallow neural networks. We also conduct experiment with linear SVM to see how well linear classifier is for our task.

Figure 8: The restaurant layout

5.3. Virtual Reality Scenario Description

We investigate a small scale scenario in a restaurant as shown in Figure 8. The restaurant layout is about 30 by 30 meters. It is broken down into 11 waypoints (shown as red circles with numbers) with 4 starting or ending points (shown as blue boxes with texts), namely the entrance, the wine cabinet, the washroom, and the table. The white color regions are the walkable regions, the green lines are the routes, and the black regions are the walls, tables and chairs.

We carried out three experiment sessions for each player, with each session starting and ending with different positions in the restaurant layout. The player is to move from start position to end position, and during the navigation, the player will encounter different junctions. When the player reaches a specific junction, a crowd of random size will be placed on the shorter path of the junction to the destination and there will be no crowd on the longer path. The player will select one of the two paths towards the destination. The details of the logic of the crowd generation process are shown in Figures 9, 10, and 11 as route decision trees.

Scene 1

- From entrance(left) to bar counter(top right)

Figure 9: Route decision tree for paths from entrance to bar counter

Before experiment starts, we will brief the players about the objective and a printed map will be given to each player to familiarize with the VR restaurant layout. Before a session starts, there is a random exploration period where each player will randomly walk in the VR layout using his controller to get to know how to operate the VR equipment and understand more about the actual 3D model of the restaurant. Then, session 1 will start and the player will move according to the paths in Scene 1 as shown in Figure 9. After that, the player will complete sessions 2 and 3 (do path findings) for Scene 2 and 3 as shown in Figures 10 and 11 respectively. At the end of experiment, each player will have to complete a survey to check whether he experiences any VR sickness such as headache or nauseousness, and to answer open ended questions to show his/her understanding of the experiment.

In Figure 9, the player starts at the entrance (purple box with green label 0). He/She will encounter the first junction that splits between node 0 and node 8. A crowd of random size will be placed on node 0 which is the shorter path towards the destination (bar counter, purple box with green label 2). If the player selects node 0, he/she will traverse node 1 and node 2, then reaches

another junction that splits between nodes 11 and 3. A crowd of random size will be placed on node 3. We assume that the player will not traverse nodes that he/she has visited before by moving in the opposite direction of the destination.
405 If the player selects node 3, then he will go to the destination at the bar counter directly as it is in front of the player. If the player selects node 11, then another random size crowd will be placed on node 10. The player has the choice of taking either node 9 or node 10 to the destination. After making the choice, we assume
410 that the player will eventually take node 5 and node 4 to the destination.

If the player selects node 8 at the entrance, he/she will traverse node 7 and reach the junction that splits between nodes 9 and 6. Again he/she has two paths to choose from and a crowd of random size will be placed at node 9. If the player selects node 9, he will go directly to the destination and there
415 will not be another crowd generated on the path. If the player selects node 6, a crowd of random size will be placed on node 5. After that, the player can either traverse nodes 10, 11 and 3 to the destination or nodes 5 and 4 to the destination. Note that a random crowd will be generated only when the player has reached specific junctions (i.e., the entrance 0, node 7, node 6, node 2, and
420 node 11). The dots inside a circle are the stationary crowds at the waypoint. These crowds of random sizes will be generated at the start of each virtual reality experiment. Note that each junction has two route choices with an OD pair (O is the junction and D is the endpoint). For each OD pair, we record the binary choice. Through this scene, we extracted at most 5 OD pairs at the
425 specific junctions and 5 binary route choices at these OD pairs.

Figure 10 shows the route decision tree for paths in the second scene. The logic is the same as Scene 1. The player will start at the washroom (purple box with green label 3) and end at the entrance. During the walk towards the destination, the player may encounter junctions the washroom, node 3, node
430 11, node 5, and node 6. Crowd of random size will be placed on node 5, node 2, node 9, node 6, and node 7 respectively. Through this scene, we extracted at most 5 OD pairs at the specific junctions and 5 binary route choices at these OD pairs.

Scene 2

- From washroom(bottom right) to entrance(left)

Figure 10: Route decision tree for paths from washroom to entrance

Scene 3

- From Table(middle) to bar counter(top right)

Figure 11: Route decision tree for paths from table to bar counter

(a) Screenshot 1

(b) Screenshot 2

(c) Screenshot 3

(d) Screenshot locations

Figure 12: (a) to (c) Sample screenshots and (d) their locations

Figure 11 shows route decision tree for paths in the last scene. The logic is
 435 also the same as Scene 1. The player will start at the table (purple box with
 green label 1) and end at bar counter. During the walk towards the destination,
 the player may encounter junctions the table, node 9, node 11, and node 10.
 Crowd of random size will be placed on node 11, node 6, node 3, and node
 5 respectively. Note that at junction node 11, there are three paths to the
 440 destination by trasvering either node 2, node 3, or node 10. Since we assume
 players will not move in the opposite direction of the destination, node 2 will
 not be considered as it leads to a significantly longer path. Through this scene,
 we extracted at most 4 OD pairs at the specific junctions and 4 binary route
 choices at these OD pairs. We carried out experiment on 6 players. Each player
 445 played a scenario two or three times. In total 126 data points were generated.

Figure 12 shows some examples of what the player sees when he/she navi-

gates through the VR environment. The crowds of random sizes are generated at specific locations as shown in the examples. The player is able to see the green lines on the floor and the destination indicated by the red arrow pointing downward. The VR environment is realistically created with sofas, tables, wine cabinet, and many other objects. The player will select two paths at specific junctions (e.g, at screenshot 1, the player will select to turn left to move through the crowd or turn right, taking a longer path to avoid the crowd). The descriptors of the paths such as distance, turn distance, and crowd size are used to learn a predictor to decide whether the pedestrian will take the shorter path or the longer one. This predictor can be plugged into a pedestrian simulation to decide which path to take given a new set of binary paths and crowd sizes.

5.4. Evaluation of Crowd Avoidance Route Choice Model

We trained multiple SVM with different kernels (linear, radial basis function (RBF), sigmoid, and quadratic kernels). The four features (distance, turn distance, angle from destination, and crowd density) of a path is further processed by taking the differences and means of the features across the two paths as shown in Table 1. The differences and means are the inputs, x , to the SVM and the output, y , of the SVM is 1 if the shorter path is selected else it is 0. Note that the inputs, x_4 to x_7 , are normalized by their standard derivation; whereas the inputs, x_0 to x_3 , are normalized by their maximums and minimums. Input-output pairs, (x, y) , of every 13 out of 14 OD pairs are used for training and the last OD pair is used for testing.

The Table 2 shows the experiment results of using different kernel SVMs against different input combinations. By using distance x_4 and density x_7 , we can achieve the best results. Without distance, using only turn angle x_5 , angle from destination x_6 , and density x_7 , we can also achieve moderate results. Without density x_7 , the distribution of selecting shorter path and longer one overlaps by a large percentage, and therefore using inputs without density produces poor results.

Figure 13 shows all pairwise scatter plots of input 4 to 7. We can see that

f0	Path 1 distance
f1	Path 2 distance
f2	Path 1 turn distance
f3	Path 2 turn distance
f4	Path 1 angle from destination
f5	Path 2 angle from destination
f6	Path 1 density
f7	Path 2 density

Features of a path

(a) Features used by our model

x0	(f0+f1)/2
x1	(f2+f3)/2
x2	(f4+f5)/2
x3	(f6+f7)/2
x4	f0-f1
x5	f2-f3
x6	f4-f5
x7	f6-f7

Inputs to SVM

(b) Inputs to our SVM

Table 1: Features and inputs to our SVM

Inputs vs classifier	(x0,x1,x2,x3,x4,x5,x6,x7)	(x4,x5,x6,x7)	(x4,x7)	(x5,x7)	(x6,x7)	(x0,x1,x2,x4,x5,x6)
Linear SVM	0.6015625	0.6171875	0.71875	0.65625	0.671875	0.4921875
Sigmoid SVM	0.515625	0.5234375	0.6640625	0.3515625	0.3671875	0.484375
Quadratic SVM	0.6640625	0.5859375	0.6875	0.6484375	0.65625	0.4765625
RBF SVM	0.703125	0.6953125	0.640625	0.5703125	0.703125	0.6015625

Accuracy for always taking shortest path is 0.375

Table 2: Route choice classification accuracies

distance $x4$ and density $x7$ are highly linear discriminative (see subplot 13c). But, the decision region between angle from destination $x6$ and density $x7$ is not so linear (see subplot 13f).

6. Conclusion

We have developed a data-driven path planning model for crowd modeling and simulation focusing on the OD popularity and route choice learning. The model can be implemented in a crowd simulation 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.

(a) Scatter plot of x4 and x5

(b) Scatter plot of x4 and x6

(c) Scatter plot of x4 and x7

(d) Scatter plot of x5 and x6

(e) Scatter plot of x5 and x7

(f) Scatter plot of x6 and x7

Figure 13: Scatter plots of our inputs to SVM (Blue: Shortest path is selected, Orange: Longer path is selected. The points are drawn with transparency ($\alpha=0.1$) so that we can visualize the overlapping points. The inputs are normalized by their minimums, maximums and standard derivations (see text for details))

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 marshal on the ground.

The assumption we make in the video scenario is that as density increases uniformly, people’s path planning is not affected much by the density increment, but still by space syntax (layout). So, one imperfection in this data-driven 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. This leads us to develop further on our route choice model by considering the crowdedness as a dynamic factor.

To quantify the impact of such a parameter, we carefully designed VR experiments to reproduce similar scenarios and invite participants to make their choices in the virtual environment with carefully deployed virtual crowds. Their choices are collected to correlate their route choices with different synthesized crowdedness. For large distance, turn distance and angle from destination differences between the two paths, the shorter path is always selected. For small differences, density plays an important role. Some regions of inputs are random as there are some people who choose one path and others who choose another path. Linear and RBF kernel SVMs perform well for route choice classification considering both space syntax features and the crowdedness feature. With this density aware route choice model, we can simulate pedestrian navigation behavior with various densities more realistically.

In our future work, we may generalize our model to the cases where there are moving crowds with crowd following behavior. We can have multiple users to take part in the experiments at the same time. We can also have larger layout with more junctions, routes, and OD pairs.

References

- [1] R. Wang, Y. Zhang, H. Yue, Developing a new design method avoiding latent congestion danger in urban rail transit station, *Transportation Research Procedia* 25 (2017) 4083–4099.
- 520 [2] H. Wang, L. Yu, S. Qin, Simulation and optimization of passenger flow line in lanzhou west railway station, in: *Scientific And Technical Conference Transport Systems Theory And Practice*, Springer, 2017, pp. 61–73.
- [3] N. Molyneaux, R. Scarinci, M. Bierlaire, Pedestrian management strategies for improving flow dynamics in transportation hubs, in: *17th Swiss*
525 *Transport Research Conference (STRC)*, no. EPFL-TALK-229186, 2017.
- [4] L. D. Vanumu, K. R. Rao, G. Tiwari, Fundamental diagrams of pedestrian flow characteristics: a review, *European transport research review* 9 (4) (2017) 49.
- [5] A. M. Rao, K. R. Rao, Measuring urban traffic congestion-a review., *Inter-*
530 *national Journal for Traffic & Transport Engineering* 2 (4).
- [6] J. J. Fruin, Pedestrian planning and design, Tech. rep. (1971).
- [7] C. G. Prato, Route choice modeling: past, present and future research directions, *Journal of Choice Modelling* 2 (1) (2009) 65 – 100.
- [8] S. K. Tan, Visual detection and crowd density modeling of pedestrians, Ph.D. thesis, School of Computer Science and Engineering, NTU,
535 <http://hdl.handle.net/10356/72746> (2017).
- [9] D. Helbing, P. Molnár, Social force model for pedestrian dynamics, *Physical Review E* (1995) 4282–4286.
- [10] P. Fiorini, Z. Shiller, Motion planning in dynamic environments using velocity obstacles, *The International Journal of Robotics Research* 17 (7)
540 (1998) 760–772.

- [11] J. Shi, C. Tomasi, Good features to track, in: Computer Vision and Pattern Recognition, 1994, pp. 593–600. doi:10.1109/CVPR.1994.323794.
- [12] S. K. Tan, N. Hu, W. Cai, Data-driven agent-based simulation for pedestrian capacity analysis, in: International Conference on Computational Science, Springer, 2018, pp. 103–116.
- [13] Y. Asakura, E. Hato, M. Kashiwadani, Origin-destination matrices estimation model using automatic vehicle identification data and its application to the han-shin expressway network, Transportation 27 (4) (2000) 419–438.
- 550 [14] J. Zhong, W. Cai, L. Luo, H. Yin, Learning behavior patterns from video: A data-driven framework for agent-based crowd modeling, in: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, 2015, pp. 801–809.
- [15] J. Zhong, W. Cai, M. Lees, L. Luo, Automatic model construction for the behavior of human crowds, Applied Soft Computing 56 (2017) 368–378.
- 555 [16] P. Charalambous, I. Karamouzas, S. J. Guy, Y. Chrysanthou, A data-driven framework for visual crowd analysis, in: Computer Graphics Forum, Vol. 33, Wiley Online Library, 2014, pp. 41–50.
- [17] M. Zhao, S. J. Turner, W. Cai, A data-driven crowd simulation model based on clustering and classification, in: IEEE/ACM 17th International Symposium on Distributed Simulation and Real Time Applications, IEEE, 2013, pp. 125–134.
- 560 [18] D. Wolinski, S. J. Guy, A.-H. Olivier, M. Lin, D. Manocha, J. Pettr, Parameter estimation and comparative evaluation of crowd simulations, in: Computer Graphics Forum, Vol. 33, Wiley Online Library, 2014, pp. 303–312.
- 565 [19] S. J. Guy, J. Van Den Berg, W. Liu, R. Lau, M. C. Lin, D. Manocha, A statistical similarity measure for aggregate crowd dynamics, ACM Transactions on Graphics (TOG) 31 (6) (2012) 190.

- 570 [20] H. Wang, J. Ondřej, C. O’Sullivan, Trending paths: A new semantic-level metric for comparing simulated and real crowd data, *IEEE transactions on visualization and computer graphics* 23 (5) (2017) 1454–1464.
- [21] S. K. Tan, T.-J. Cham, J. Wu, Steerable second order intensity features for pedestrian detection, in: 3rd IAPR Asian Conference on Pattern Recognition (ACPR), IEEE, 2015, pp. 311–315.
- 575 [22] H. Wang, C. O’Sullivan, Globally continuous and non-markovian crowd activity analysis from videos, in: *European Conference on Computer Vision*, Springer, 2016, pp. 527–544.
- [23] S. Das, P. N. Suganthan, Differential evolution: A survey of the state-of-the-art, *IEEE transactions on evolutionary computation* 15 (1) (2011) 4–31.
- 580 [24] W. Yi, J. Zhong, S. Tan, W. Cai, N. Hu, Surrogate assisted calibration framework for crowd model calibration, in: *Winter Simulation Conference (WSC)*, IEEE, 2017, pp. 1216–1227.
- 585 [25] J. Zhong, N. Hu, W. Cai, M. Lees, L. Luo, Density-based evolutionary framework for crowd model calibration, *Journal of Computational Science* 6 (2015) 11–22.
- [26] A.-H. Olivier, J. Bruneau, G. Cirio, J. Pettré, A virtual reality platform to study crowd behaviors, *Transportation Research Procedia* 2 (2014) 114–122.
- 590 [27] B. Zhou, X. Wang, X. Tang, Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents, in: *Computer Vision and Pattern Recognition*, IEEE, 2012, pp. 2871–2878.
- [28] S. Fadel, S. Ghoniemy, M. Abdallah, H. A. Sorra, A. Ashour, A. Ansary, Investigating the effect of different kernel functions on the performance of svm for recognizing arabic characters, *IJACSA) International Journal of Advanced Computer Science and Applications* 7 (1) (2016) 446–450.
- 595