

PAPER: Interdisciplinary statistical mechanics

Toward learning dynamic origin-destination matrices from crowd density heatmaps

Marion Gödel^{1,2}, Daniel Lehmberg^{1,2},
Rebecca Brydon¹, Ernst Bosina³ and Gerta Köster¹

¹ Department of Computer Science and Mathematics, Hochschule München
University of Applied Sciences, Lothstrasse 64, 80335 Munich, Germany

² Department of Informatics, Technical University of Munich,
Boltzmannstrasse 6, 85748 Garching, Germany

³ Swiss Federal Railways SBB AG, Gardistrasse 2, 3000 Bern 65,
Switzerland

E-mail: marion.goedel@hm.edu

Received 17 December 2021

Accepted for publication 24 March 2022

Published 12 May 2022

Online at stacks.iop.org/JSTAT/2022/053401

<https://doi.org/10.1088/1742-5468/ac6255>



Abstract. Knowing the origins and destinations of pedestrians' paths is key to the initialization of crowd simulations. Unfortunately, they are difficult to measure in the real world. This is one major challenge for live predictions during events such as festivals, soccer games, protest marches, and many others. Sensor data can be used to feed real-world observations into simulations in real-time. As input data for this study, we use density heatmaps generated from real-world trajectory data obtained from stereo sensors. Density information is compact, of constant size, and in general easier to obtain than e.g., individual trajectories. Therefore, the information limitation improves the applicability to other scenarios. We include the absolute pedestrian trip counts from origins to destinations during a brief time interval in an OD matrix, including unknown destinations due to sensor errors. Our goal is to estimate these OD matrices from a series of density heatmaps for the same interval. For this, we compute the ground truth

*Author to whom any correspondence should be addressed.



Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](https://creativecommons.org/licenses/by/4.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

C.2. Number of features for split23
 C.3. Maximum depth of the trees23
Appendix D. Overlapping data set24
References25

1. Introduction

Crowd simulations are an important tool to enhance safety and security at events. They are cheap to perform when compared to experiments and field observations, and they are simple to modify to study different setups. There are several models available today for microscopic crowd simulations. Common models are based on differential equations, such as social force models [1, 2], rule-based models, such as cellular automata [3], or utility-based models, such as the optimal steps model [4, 5].

In most agent-based simulations, the origin and destination of agents are essential for the initialization of the simulation. In addition, they have a great impact on the prediction. Values are frequently chosen today based on experience, observations from experiments or in the field, and surveys. While these initialization options are appropriate and useful in a variety of situations, they are typically static or only vary over longer time frames, at least hours.

Today, crowd simulations are not only used during the planning phase, but also for live predictions at events [6–10]. This relatively new application presents challenges: a static initialization is no longer sufficient. To perform a realistic live prediction, methods for using data from sensors as input for the simulation must be identified. The simulation must be fed continuously with reliable dynamic information on the origins and destinations of the agents.

1.1. Goals

We propose a parameter learning approach based on real-world data to determine the number of pedestrians moving from each origin to each destination. This methodology can be integrated into an online learning setup for a live prediction. Our data set contains pedestrian trajectories in a train station overpass with stairs, escalators, and elevators that lead to the platforms provided by Swiss Federal Railways (SBB).

In most setups, real-world trajectory data is hard to obtain, even when video footage of crowds is available, since extracting trajectories from video material is still a non-trivial task. Density heatmaps, on the other hand, can be obtained from a variety of sensor types, such as video data [11]. In addition, the density heatmaps are of constant size for a given observation area and resolution. This is necessary for the statistical learning models. As a result, instead of using the full trajectories, we derive crowd density heatmaps from the trajectories to increase the likelihood of application.

J. Stat. Mech. (2022) 053401

The information about the origins and destinations of the pedestrians is encoded in an origin-destination (OD) matrix that, in addition, contains information about the popular paths. This is a standard format for pedestrian dynamics and traffic simulation. It can be used as a Markov chain model for direct prediction as well as to initialize simulations. We are interested in the latter case, particularly in the context of live predictions.

1.2. State-of-the-art OD matrix estimation

Real-time event prediction is a new challenge in the pedestrian dynamics community. Therefore, several issues need to be addressed. Most importantly, code acceleration and initialization. At a minimum, the algorithms must be real-time capable. To improve pedestrian safety, the computation of predictions has to be fast enough to allow enough time to take action. Static initialization is no longer sufficient. Instead, the current state must be translated into simulation inputs and fed to the simulation. The problems associated with real-time capability, as well as approaches to solving them, are described in [12]. Here we focus on initialization, keeping in mind the computational cost.

First, we take a look at studies that aim to set up systems for real-time prediction. Kemloh Wagoum *et al* propose a real-time evacuation assistant for evacuating 50 000 pedestrians from a stadium [8]. An automated pedestrian count is fed into the simulation. The paper's emphasis, however, is on parallelizing and speeding up the algorithm. Baqui and Löhner present a framework for analyzing crowd density in Hajj video footage [6]. A machine learning model is used to estimate the density from headcounts. The estimation's goal is to initialize a microscopic crowd simulation.

Next, we focus on the literature on the estimation of OD matrices. These publications focus on OD matrix estimation without providing a framework for real-time predictions. OD matrix estimation is an established research topic in traffic simulations [13, 14]. Pitombeira Neto *et al* present a review in [15]. Static OD matrices are often sufficient for automobile traffic. However, dynamic OD matrices are required for crowd simulation because we observe more fluctuations and pedestrians have more degrees of freedom in their movement. The first step in estimating OD matrices is to specify the origin and destination of the agents. We define them based on the trajectory data in the overpass scenario. This, however, is not always possible. Khan *et al* use an unsupervised hierarchical clustering algorithm to determine the locations of origins and destinations using short trajectories, known as tracklets, automatically extracted from the video footage [16]. Their approach is applied to six scenarios: an airport, the Hajj, a train station, an escalator, a university campus, and a gallery in downtown Milan.

In [17], the dynamic origin-destination matrices are only an intermediate result to validate a crowd model based on points of interest. Pouke *et al* derive city-level OD matrices from Wi-Fi mobility traces collected in downtown Oulu, Finland. Using a convolutional neural network (CNN), Li *et al* estimate OD matrices from RGB-D images for low visibility emergencies [18]. RGB-D images are created by combining an RGB image with the corresponding depth image. CNNs are appropriate for this task because

they are trained on images directly. The ground truth is obtained by manually labeling pedestrians. The study is conducted using RGB-D cameras installed in the hallways of a university building. Chan *et al* employ a two-stage programming model, a commonly used approach for estimating the OD matrix in automobile traffic simulation [19]. The inputs to the method are pedestrian counts. The two-stage programming model is demonstrated on an abstract, large-scale network.

Pedestrian traffic in stations or even in a citywide transit system is of greater interest because these areas have a larger number of potential origins and destinations. Wong and Tong estimate dynamic OD matrices for the entire transportation system using observed ridership from Hong Kong [20]. They employ an entropy-based approach. To predict pedestrian flows within Takatsuki Station in Japan, Ahn *et al* estimate a static OD matrix [21]. They estimate the OD matrix using Kawakami's gravity model. The estimation is based on manually observed traffic counts at the station. Bauer calculates daily averaged OD matrices using pedestrian counts collected by sensors at entrance and exit points [22]. This data comes from an Austrian shopping mall. The data is divided into two sets, training and validation, and the performance is measured using the R^2 score. A generalized method of moments is used to estimate the OD matrices. Hänseler *et al* estimate dynamic OD matrices at Lausanne station based on tracking data, travel surveys, and train schedules [23]. While they have a similar goal, we estimate OD matrices using only relatively simple density data rather than combining different types of data. This information restriction makes it much easier to collect data in a complex scenario, such as a train station, reliably and continuously. As a result, its applicability improves.

1.3. Research question

The research question of this study is: can we learn dynamic OD matrices? More precisely, can we estimate OD matrices for a short time interval based on a series of density heatmaps? We perform a feasibility study that addresses this research question. Our goal is to find out if it is possible to predict the OD matrix for a brief time interval given a short series of density maps. It is not in our interest to find the model with the lowest error.

Previously, we investigated how to predict the distribution of a pedestrian flow across three different destinations [24]. The crowd simulation tool Vadere [25] was used to simulate the surrogate input data from which we obtained the heatmaps. Based on this, we incorporated experimental trajectory data from the Jülich research center [26] to predict the origin distribution of three pedestrian flows that all lead to the same destination [27]. In both cases, the estimation of the distribution of agents or pedestrians between origins and destinations is highly accurate. However, the studies were based on either artificial data or controlled experiments, both with unidirectional flows and limited variation in behaviors. While the results are encouraging, their informative value is limited, and one hesitates to generalize to real-world data. Therefore, we now test ideas and methods on real-world data to see if it is possible to predict where people come from and where they go. In a train station, we investigate pedestrian flow through an overpass

with stairs leading to platforms and the station hall. Since the flow is multidirectional, predicting the distribution of agents only among the origins or destinations is no longer sufficient. Instead, we forecast entire OD matrices.

2. Preprocessing of trajectory data

In this section, we describe the train station overpass in which the pedestrian trajectories are recorded. We provide a brief analysis of the observed speeds and the pedestrian count. Then, we describe the calculation of the density heatmaps and the ground truth OD matrices from the dataset.

2.1. Scenario: train station overpass

For this study, we use pedestrian trajectories recorded on three days of September 2019 by SBB in an overpass (of approximately $60 \text{ m} \times 20 \text{ m}$) in a train station with exits to several platforms and the station hall (cf figure 1). Xovis' stereo sensors were used to generate the trajectories. A collection of these sensors is installed in the ceiling. Each pedestrian's location is determined and tracked across the sensor array. Tracks of three-dimensional positions with a temporal resolution of 100 milliseconds are contained in the dataset. For this study, height data is omitted. The data contains some flaws, such as time gaps of up to three seconds during which the positions are missing. Furthermore, whenever the sensor loses track of a pedestrian, a new measurement is initiated. Therefore, some trajectories may be broken into multiple parts with different pedestrian ids.

The density heatmaps and the ground truth OD matrices, used as input and output for the estimation, are generated from the raw trajectory data. The density is calculated using trajectory snippets within a time interval of fixed length, $\Delta t = 10 \text{ s}$. This decision is based on the rough calculation below: pedestrians move an average of eleven meters in ten seconds, which is a substantial distance in the overpass but shorter than the OD links. Time intervals with fewer than ten pedestrians in the overpass are not taken into account. The trip counts for the true OD relationships are derived directly from the trajectory data as well. Note that we ignore measurement errors in the trajectories. We would also like to point out that any method for obtaining density maps from video footage entails errors that we cannot account for here. In our case, the raw data stems from stereo sensors installed on an indoor ceiling, hence their setup is optimized for pedestrian tracking. In most applications, this is not the case and measurement error may compromise the process we propose in this contribution.

2.2. Analysis of trajectory data

The dataset consists of three working days. Consequently, we most likely observe commuters and travelers within all age ranges. However, the sensor data does not include information about the observed population's age, fitness, or other characteristics. We

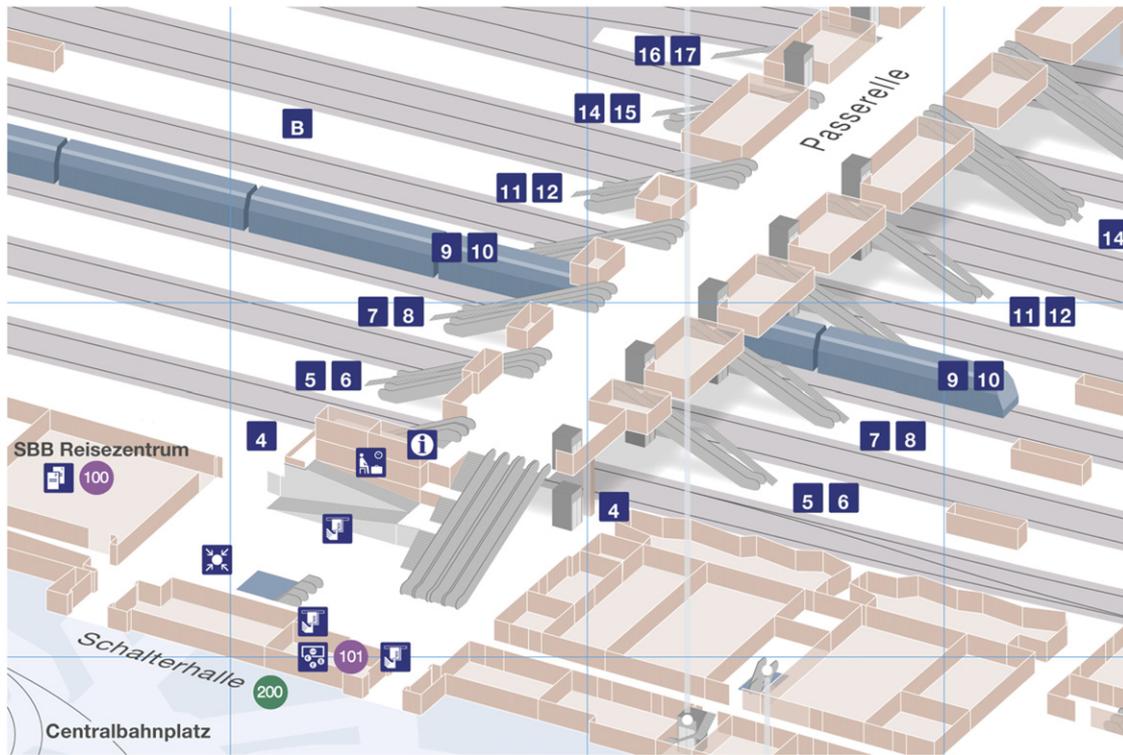


Figure 1. Plan of the train station overpass (accessed on Dec 8, 2021, © SBB/CFF/FSS 07/2021, Geodata: © OpenStreetMap contributors, © swisstopo (5704003351))⁴.

analyze the speed of the pedestrians to gain some indirect insight. Figure 2 depicts the asymmetric distribution of walking speeds. The speed is measured over the entire time a pedestrian is inside the measurement area, including periods when the pedestrian is stationary. This could be the reason for several slow pedestrians with speeds below 1 m s^{-1} . The speeds are within the value range of 0.00 m s^{-1} to 2.50 m s^{-1} , with a mean of 1.13 m s^{-1} , and a mode of 1.17 m s^{-1} and standard deviation of 0.37 m s^{-1} . This is in accordance with earlier measurements [28–30]. The results are also in line with those by Davidich and Köster [31] who measured a mode of 1.04 m s^{-1} and standard deviation of 0.51 m s^{-1} during rush hours at a train station.

Figure 3 shows how the evolution of the number of pedestrians inside the observation area varies over one day for the three weekdays of our study. We observe a similar behavior of pedestrian counts for all three days. This is most likely due to commuters: the number of pedestrians increases during rush hours (5–9 a.m. and 4–7 p.m.), while it decreases during the day and at night. Alighting passengers from arriving trains cause short-term peaks. We presume that the OD matrices require dynamic initialization due to the variation in the trend of the number of pedestrians per day.

⁴<https://sbb.ch/content/dam/infrastruktur/trafimage/bahnhofplaene/plan-basel-sbb-plakat.pdf>.

Toward learning dynamic origin-destination matrices from crowd density heatmaps

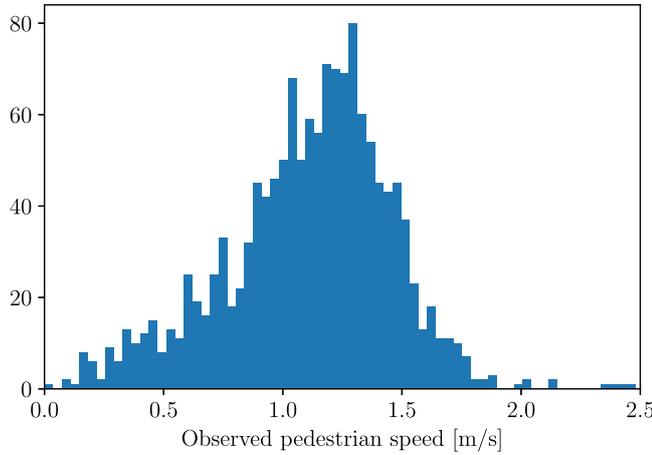


Figure 2. Observed speeds of all pedestrians within the overpass. The speed is measured over the complete time a pedestrian remains inside the measurement area, including stationary periods. Outliers over 2.5 m s^{-1} have been removed from the figure.

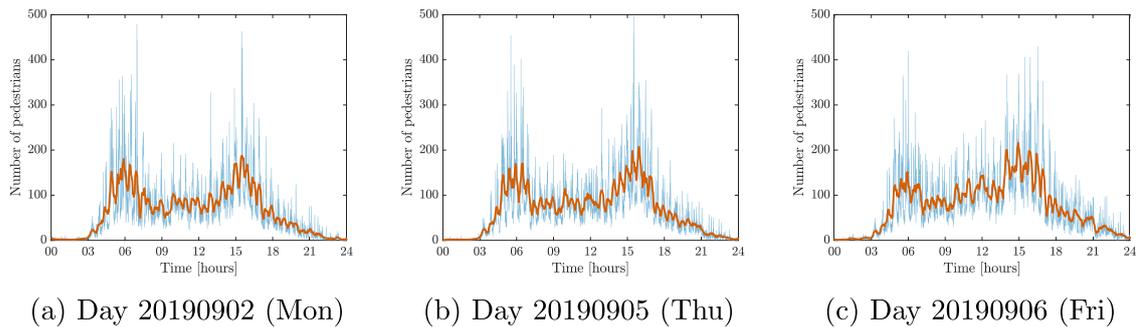


Figure 3. Evolution of the number of pedestrians in the overpass over the three separate days of the data set, Monday, Thursday, and Friday. The instantaneous pedestrian count is evaluated every ten seconds. In addition, a moving average filter with a span of 120 (≈ 20 minutes) shows the trend for the pedestrian count.

2.3. Calculation of density heatmaps

For each time interval of $\Delta t = 10 \text{ s}$, we create a series of five density heatmaps at equidistant time steps ($t_0 = 0 \text{ s}$, $t_1 = 2 \text{ s}$, \dots , $t_4 = 8 \text{ s}$). At each time step, the density at each position in the overpass is computed with a Gaussian filter [4]. That means, each pedestrian within the observation area and within the time interval is represented by a Gaussian bell. The observation area is divided into a regular grid of 0.5 m width per cell. The contributions of all nearby pedestrians' Gaussian representations are added up for each cell. As a result, we have five density heatmaps with 108×38 values each.

Figure 4 depicts an example of a series of density heatmaps for a 10 s time interval. The corresponding trajectory snippets are shown in figure 4(a). We can see from the density evolution, that there is a multidirectional movement within the observation area.

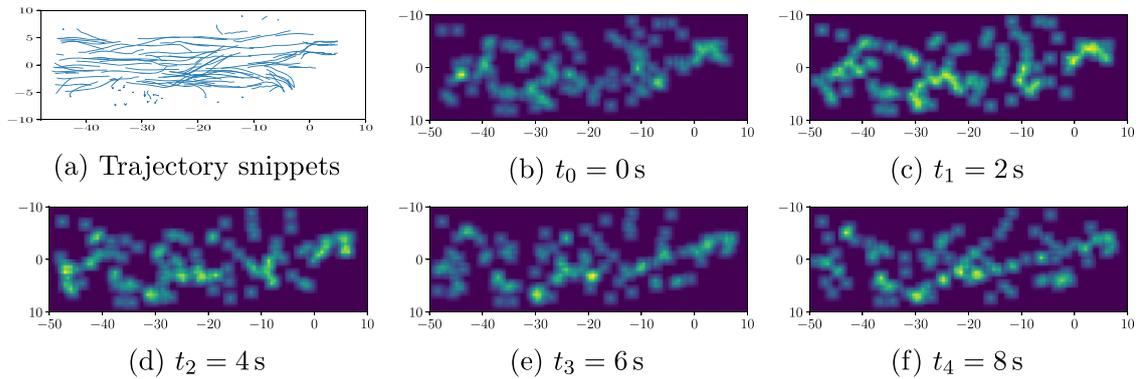


Figure 4. From trajectory snippets (a) to a series of five density heatmaps (b)–(f), both for a time interval of 10 s. The color encodes the pedestrian density from blue to yellow (low to high). Coordinates (x, y) indicate the positions in the train station overpass.

Based on the observed average speed of the pedestrian, a typical pedestrian moves about 4.5 cells of the density grid between two successive heatmaps for a sampling frequency of 0.5 Hz.

2.4. Decomposition of density heatmap series

The density heatmap matrices are sparse as can be seen in figure 4. Thus, we choose to reduce their dimensionality by a principal component analysis (PCA). As an additional benefit, the PCA helps to reduce the noise in the data. The PCA decomposes the density heatmaps into

$$X = WZ^T \approx W_{\hat{n}}Z_{\hat{n}}^T \quad (1)$$

where $X \in \mathbb{R}^{N_t \times n}$ contains a series of N_t centered flattened density heatmap samples each with $n = 108 \times 38 = 4104$ features, W contains the principal components of X (or eigenvalues of $X^T X$), and Z contains the right singular vectors of X (or eigenvectors of $X^T X$). The magnitude of the singular values of X is used to sort the singular vectors in Z in descending order. The reduced matrix has dimensions $N_t \times \hat{n}$ because only the first \hat{n} components are kept. The routines save the matrix $Z_{\hat{n}}$. It is necessary for the reconstruction of X or its approximation \hat{X} . X can only be fully recovered if $\hat{n} = n$. For the case that $\hat{n} < n$, the reconstruction approximates X .

Figure 5 shows the explained variance in the reconstructed density heatmaps over the number of components. It indicates how well the components explain the original data. We decided to use 321 components to reach an explained variance of 75%. After decomposition, each density heatmap series is represented by 321 components.

Due to the stacking of the density heatmaps and the PCA decomposition, the spatial and temporal relationships in the input samples are difficult to impossible for humans to discern. For computerized processing, however, this is not a problem: the stacking of the heatmaps can be understood as time-delayed embedding. Thereby we enrich the latest density heatmap with temporal context of the preceding heatmaps. This procedure is

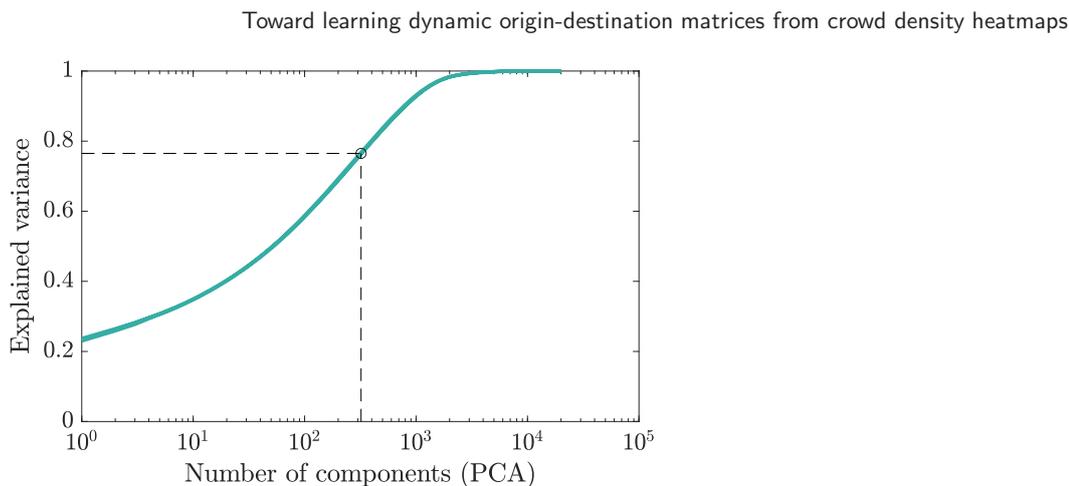


Figure 5. Explained variance of the PCA of the input samples. Each sample contains five density maps computed at equidistant steps within a time interval of ten seconds.

also often applied in methods to reconstruct the dynamics in dynamical systems (cf Takens' theorem [32]). Here, we aim to improve the mapping to the OD matrices. The input samples therefore include both the spatial and temporal context in the features. By applying the PCA decomposition, the spatial and temporal structure in the input sample space are encoded in the PCA components.

In addition to the PCA decomposition performed on the input samples, we perform a PCA on the targets, the OD matrices which follows the same principle. Instead of fixing the number of components for the target PCA, we evaluate the performance of each model for different numbers of components.

2.5. Calculation of ground truth OD matrices

By plotting the pedestrian positions of a complete day, we identify nine relevant areas in which most of the trips start or end (cf figure 6). As shown in figure 1, they correspond to escalators, stairs, and elevators that lead to the platforms (ids 1, 2, 7, 8) as well as the station hall (ids 3, 4, 5) and the remaining portion of the overpass that is not covered by the sensor (id 0). With the ids 0 to 8, we use these areas as origins and destinations.

To determine the OD matrices for a time interval Δt , each pedestrian within the observation area is mapped to one origin and one destination. If the start or endpoint of the full trajectory is outside of the identified areas, we assign an artificial origin or destination with index -1 . In this way, we can map about 65% of the pedestrians to both an origin and a destination (10% can be assigned to neither source nor destination, 11% only to a source, and 12% only to a destination). Since the positions and trajectories, and thus the densities, are affected by other pedestrians, we decided not to artificially remove people from the scenario. Please note that this means that our ground truth is not ideal, which is likely to impact our results.

Our definition of the OD matrix is based on trip counts. Each entry $a_{i,j}$ of the OD matrix $A = (a_{i,j})_{i,j \in [1,10]}$ contains the absolute number of pedestrians who traveled from

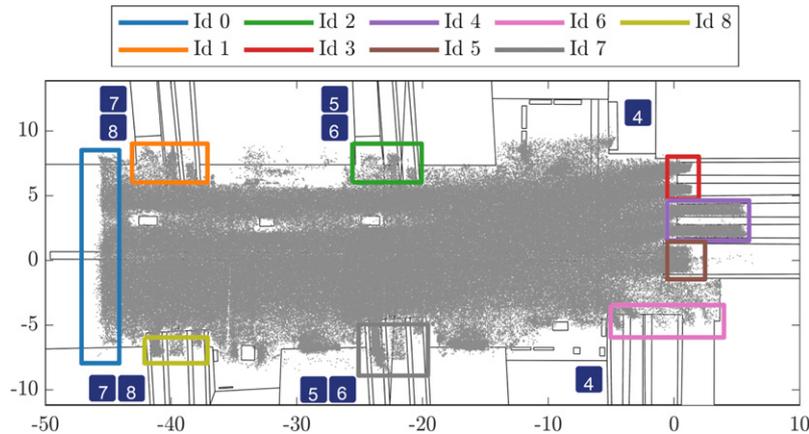


Figure 6. Pedestrian positions of a single day in the train station overpass show which areas are strongly frequented. Nine areas (ids 0–8) are identified as relevant origins and destinations: escalators, stairs, and elevators leading to the platforms (ids 1, 2, 6, 7, 8) as well as to the station hall (ids 3, 4, 5), and the remaining portion of the overpass (id 0). Coordinates (x, y) indicate the positions in the train station overpass. Blue markers show the track numbers from figure 1.

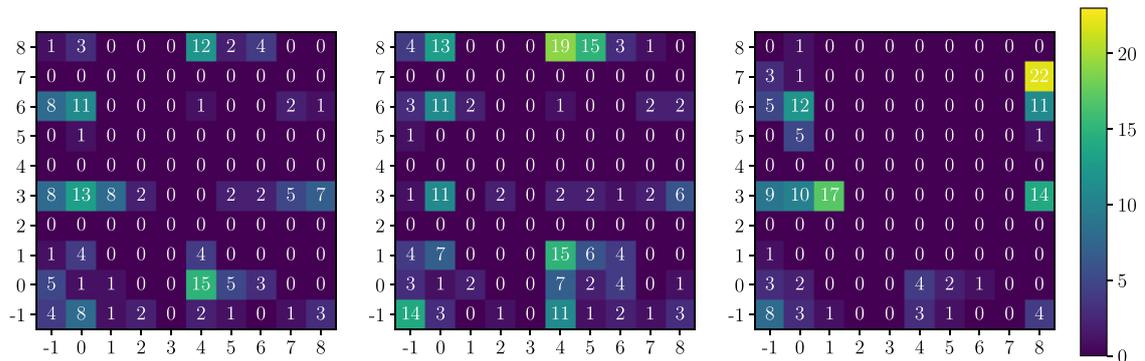


Figure 7. Exemplary OD matrices for time intervals of 10 s. Each entry contains the absolute trip count of pedestrians moving from origin i (row) to destination j (column). Pedestrians whose trajectories could not be matched with an origin or destination, are assigned -1 as origin and/or destination index.

origin i to destination j . This leads to a 10×10 OD matrix. Exemplary OD matrices can be seen in figure 7. We observe that the OD matrices are dynamic, as expected.

3. Setup of statistical learning models

We employ scikit-learn [33] routines for multivariate linear regression and random forest to estimate the OD matrices. Therefore, in our description of the methods, we stick to the machine learning terminology. The model is trained on a set of samples and targets.

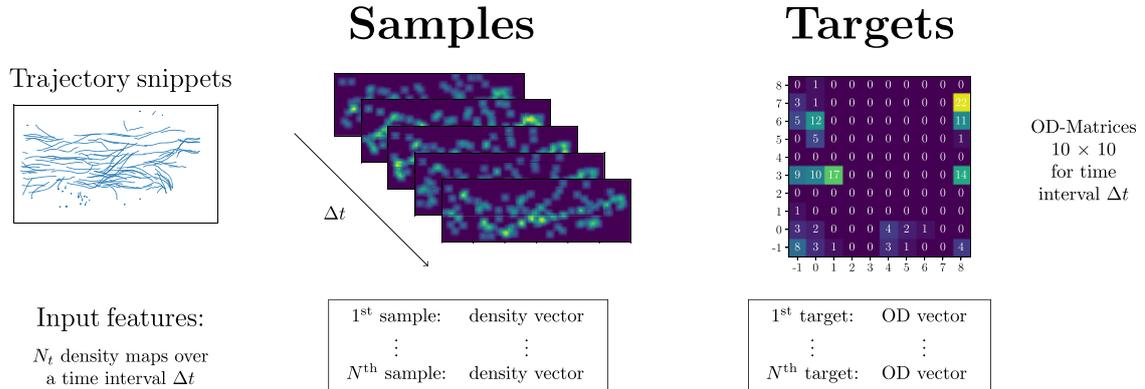


Figure 8. Processing scheme for the estimation of OD matrices from a series of density heatmaps.

The samples in our case are the density heatmaps and the targets are the ground truth OD matrices. Following training, the performance of each algorithm is evaluated on a test set that only contains samples and targets that were not included in the training. The setup is visualized in figure 8.

For the input samples, the time interval Δt is sampled with $N_t = 5$ density maps. After decomposition, each input sample contains 321 entries. These are then flattened in row-major and concatenated to an input sample of size $N_t \times 321$. Each input sample is paired with a target sample containing the OD matrix for the same observation time interval Δt . The OD matrix is flattened in row-major order which results in a vector with 100 target entries. The data set for a single day contains approximately 4000 samples. Two consecutive samples have disjoint time intervals.

In the context of real-time crowd predictions, the model is trained offline with existing data. The density heatmaps need to be calculated online from sensor data. From trajectory data, density heatmaps can be derived fast by using a discrete convolution for the Gaussian filter. Density heatmaps can be derived from a number of sensors. However, proposing methods for retrieving density heatmaps from other sensor data, analyzing the respective errors and their impact on the estimation is out of scope for this publication. Given the heatmaps, fast predictions of OD matrices are possible with a trained model.

3.1. Multivariate linear regression

We choose linear regression as the first model because it is well understood and fast. The method assumes a linear relationship between input and output. Since the output contains multiple target entries, multivariate linear regression is required. Multivariate regression is a type of supervised machine learning method. The general formulation for multivariate linear regression is

$$Y = ZB + E \tag{2}$$

where each row of the targets Y contains a single OD matrix, flattened in row-major order, each row in the samples Z contains the PCA components for a sequence of N_t density maps, (cf equation (1)). B contains the linear regression parameters that need to be determined in a least-squares sense and E contains the approximation errors. We fit a linear regression with intercept. For this purpose, Z is complemented with a vector of ones. Consequently, B holds the slope parameters and intercepts. Due to the centering of the matrices for the PCA, however, the estimated intercept is close to zero.

3.2. Random forest

One nonlinear supervised model that is easy to apply, comparably fast, explainable, and yet robust is random forest regression. In contrast to neural networks, random forests do not require setting up a complex model architecture and they depend only on a few parameters. In addition, unlike support vector machines, they are suitable for the prediction of multiple targets without training multiple models. All of these points make random forest a suitable model for our problem. Random forests are a statistical learning model developed by Breiman *et al* [34]. Each forest consists of a set of decision trees. Decision trees tend to overfit data, but by constructing an ensemble of trees and consolidating the results, overfitting can be reduced. Each tree is trained on a random subset of the data. Furthermore, for training, each tree employs a different random subset of the features. The goal is to reduce the correlation between individual trees. For the prediction, all trees are given the same input, and the average response of all trees is used as the prediction. Due to the multitude of trees, the model is very robust. Random forests only have three central parameters: the number of trees in the forest, the maximum depth of the trees, and the number of features considered in each split.

4. Performance and component analysis of statistical learning models

We investigate the performance of multivariate linear regression and random forest in predicting dynamic OD matrices using density heatmaps. The heatmaps are generated from real-world trajectory data from a train station overpass. The flow through the overpass is multidirectional. It varies throughout the day as a result of working hours and train schedules.

4.1. Performance metric: R^2 score

In the following, we employ the commonly used coefficient of determination, R^2 score, as the metric for the performance of the algorithm. This score is defined as

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^p (y_i - \hat{y}_i)^2}{\sum_{i=1}^p (y_i - \bar{y}_i)^2} \quad (3)$$

where y_i is the ground truth result with mean \bar{y}_i and \hat{y}_i is the prediction. It indicates how well the predicted values fit the ground truth values. The score has an upper limit of 1 and no lower limit. The scores for a multidimensional output are averaged with the variance of the ground truth samples.

We calculate two scores as a performance metric: the first is the component score, and the second is the OD matrix score. For the former, \hat{y}_i are \hat{n} predicted components of the PCA decomposition of the OD matrix (cf equation (1)). For the latter, we reconstruct the entire OD matrix $A = (a_{ij})_{i,j \in [1,10]}$ with absolute number of pedestrians travelling from origin i to destination j from the components. Consequently, \hat{y}_i in this case are the $p = 100$ entries of a flattened OD matrix.

4.2. Cross-validation

To test the generalization error and robustness of a statistical learning model for a set of data, k -fold cross-validation is performed. Therefore, the data is divided into k disjoint sections, known as folds. Then, $k - 1$ folds are used for training and one fold is used for validation. Instead of assessing the performance for one split of the data, the performance is assessed for each combination.

We perform three-fold cross-validation with a dataset consisting of three different weekdays. Two days are used as a training set and one day is used as a test set for each fold. The folds are not exactly equal in length because there is some variation in how many intervals are ignored due to a lack of pedestrians. With this configuration, we can determine whether the model's performance is affected by the day.

4.3. Multivariate linear regression

The output of our prediction, the OD matrix, has 100 entries. Most statistical learning models perform better with a scalar output or at least a low dimensional output. Therefore, we decided to reduce the output dimension by applying a PCA decomposition to our 100-dimensional target space. That is, we predict the OD matrices on the reduced basis of the PCA.

We observe that the component scores are superior when predicting a small number of components of the PCA while the OD matrix scores are inferior (figure 9). The reason is a trade-off between the accuracy of the prediction which is higher for a small number of targets and the accuracy of the reconstruction of the OD matrix which is higher for a larger number of predicted PCA components. We argue that the results, while not perfect, indicate that the linear regression model captures a significant portion of the OD matrix variation over the day. Figure 10 depicts the evolution of the components over time, which supports this. In figure 11, we report the absolute errors of the prediction of the first two components of the PCA. We observe only minor variations in the accuracy over the course of the day. Figures 12 and 13 evaluate the absolute errors for one-hour intervals for the first and second component, respectively. The errors are centered around 0. For the first components, the variation increases slightly during the rush hours.

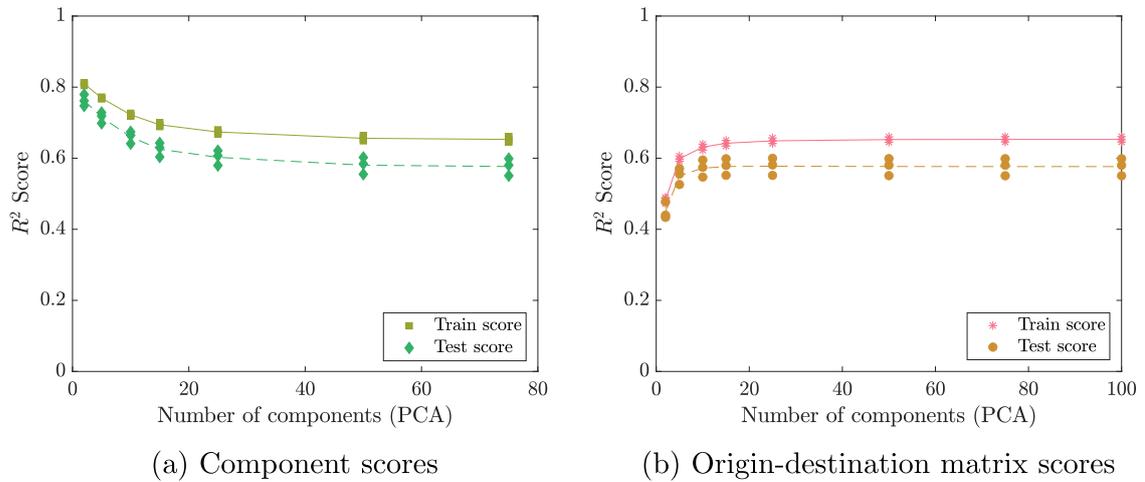


Figure 9. Performance of multivariate linear regression for the prediction of OD matrices from a series of five density heatmaps. The performance is shown with respect to the number of predicted components of the PCA of the OD matrix. The R^2 score is evaluated for the prediction of the components and the reconstructed entire OD matrix. Markers indicate the performance for each individual fold, lines show the averaged performance.

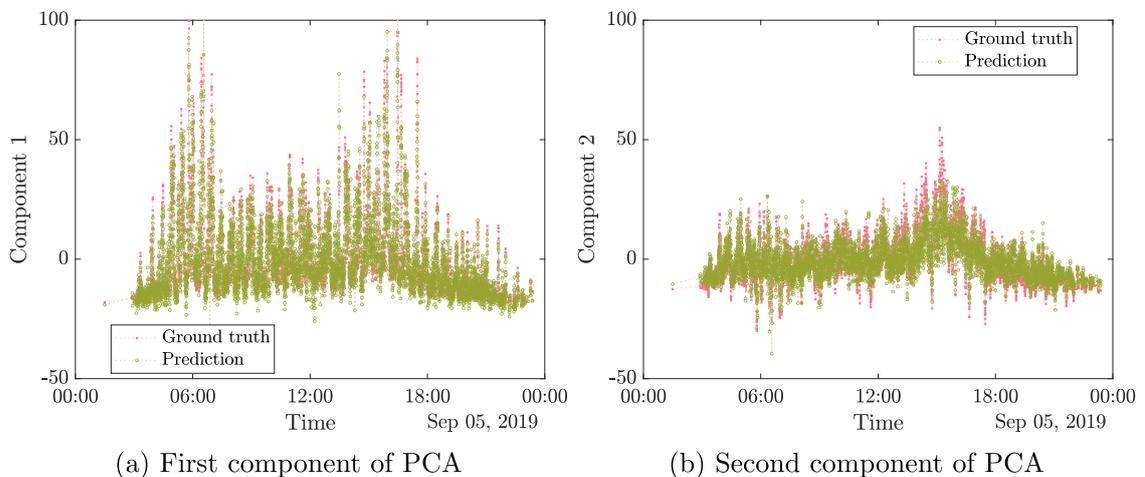


Figure 10. Predicted and ground truth components of the PCA of the OD matrices over one day.

In addition, in figure 9 the scores are shown for each fold (markers). We observe similar R^2 scores for all folds demonstrating that the model is robust against the variation between days and therefore generalizes well.

Initially, we chose five heatmaps in each sample to capture the movement of the crowd with the density heatmaps. However, we are not sure whether the linear regression model profits from the series of density heatmaps. We clarify this by comparing the

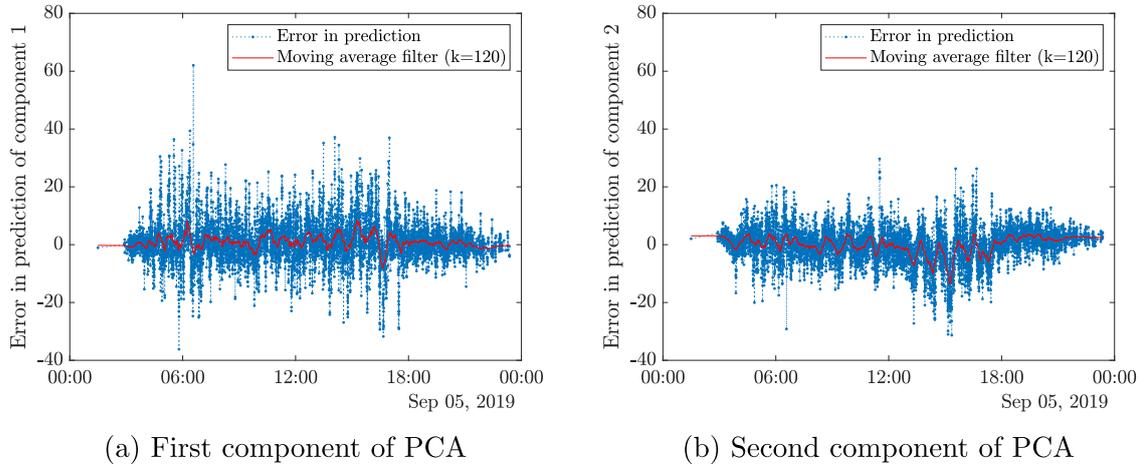


Figure 11. Difference between predicted and ground truth components of the PCA of the OD matrices over one day. A moving average filter with a lag of $k = 120$ (=20 min) is shown.

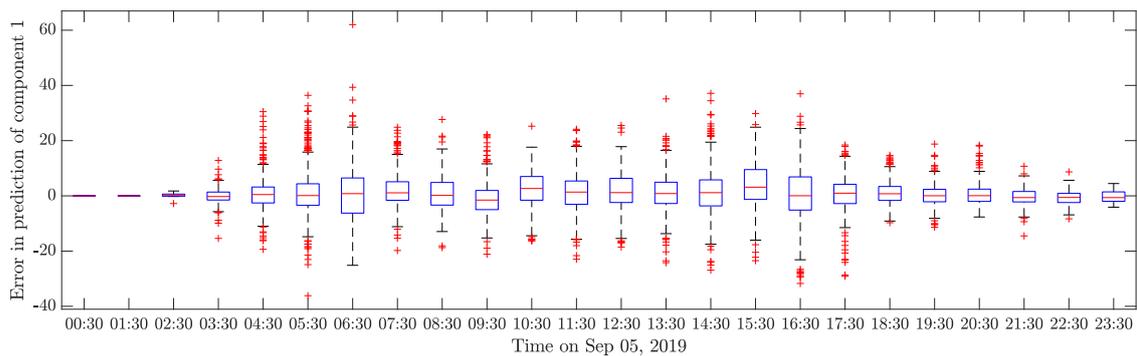


Figure 12. Difference between predicted and ground truth first component of the PCA of the OD matrices over one day. Data is binned in one-hour intervals.

performance with a single density heatmap ($N_t = 1$) to a series of $N_t = 2$ and $N_t = 5$ density heatmaps per samples. See figure 14. Again, we use the PCA decomposition for the input samples with 75% explained variance. That corresponds to 99 components for a single heatmap in each sample and 182 components for a series of two density heatmaps, respectively. We expect that using a series of density maps to predict the movements of pedestrians improves the performance. In fact, there is an increase in the R^2 score when comparing a single heatmap ($N_t = 1$) to a series of two heatmaps ($N_t = 2$). Using five heatmaps brings only a negligible improvement over two.

We also varied the number of features in the input samples by varying the number of components that we keep from the PCA decomposition of the density heatmap series. The results are shown in appendix A. They support the choice of keeping 321 components.

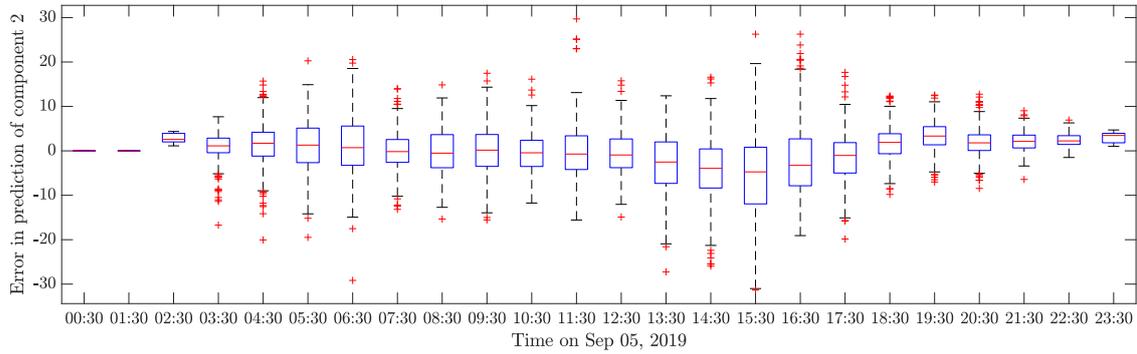


Figure 13. Difference between predicted and ground truth second component of the PCA of the OD matrices over one day. Data is binned in one-hour intervals.

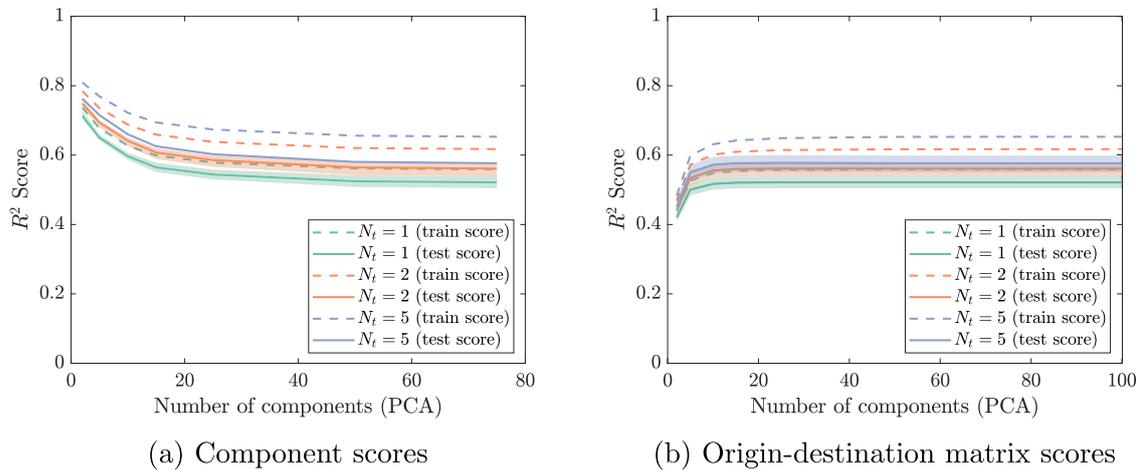


Figure 14. Average performance of multivariate linear regression with different input samples: each sample contains either a single density heatmap ($N_t = 1$) or a series of two ($N_t = 2$) or five ($N_t = 5$) density heatmaps. The performance is evaluated with respect to the number of predicted components of the PCA of the OD matrices.

Let us take a look at why the performance of the regression remains so far from an ideal R^2 score of 1. We saw in the previous sections that keeping two components of the OD matrix decomposition provides the best performance in terms of the component scores. Plotting the prediction and ground truth components shows how well the predictions fit the targets in the component domain (cf figures 15 and B1). The linear regression predictions overlap with some of the ground truth data. Nonetheless, the shapes of the data clouds vary considerably. We suspect that the overlap reflects the linear part of the relationship between input samples and target samples. This indicates that a nonlinear model is necessary to obtain a better fit.

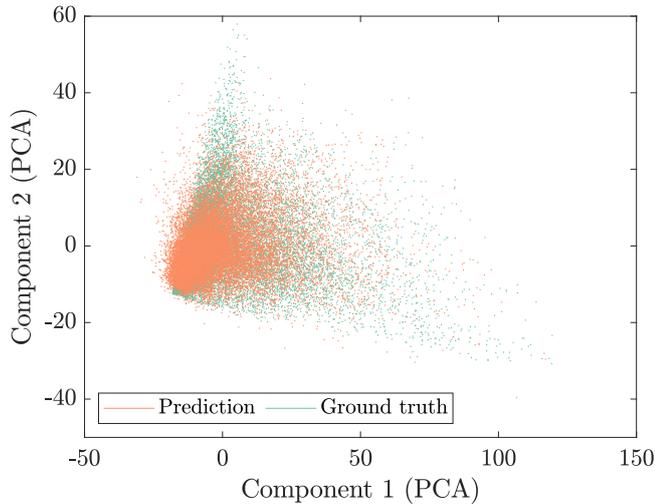


Figure 15. Scatter plot of the first and second component of the PCA of the OD matrices. The linear regression predictions of the components are shown together with the ground truth components.

4.4. Nonlinear model: random forest

We choose random forest as a nonlinear model. Random forest is a well-understood and relatively simple model with a low number of parameters. We studied the impact of the main parameters (see appendix C) and use the best configuration (number of trees: 100, maximum depth of the trees: 10, number of features considered for each split: 321 (all features)). Figure 16 shows the results. In this configuration, random forest performs slightly worse than linear regression (considering the test scores). To avoid overfitting due to a small sample size, we create a larger dataset by using overlapping time intervals (see appendix D). However, the performance of random forest does not change as the larger dataset grows larger.

Figure 17 compares the first two ground truth components and the first two predicted components of the OD matrix for linear regression and random forest. Figure B2 shows the point clouds for ground truth and prediction separately. The random forest predictions appear to fit the ground truth components much better. However, the ground truth and predicted components do not perfectly overlap. This means the random forest model does not capture all of the data set and that there is room for improvement with more advanced methods. This is encouraging news because linear regression still outperforms random forest, although random forest captures the shape better. Figure 16 depicts this. As a result, we believe that a better nonlinear model is required to improve accuracy. A brief comparison with other nonlinear models showed no significant improvements. Determining the best nonlinear model is a difficult task that is beyond the scope of this paper.

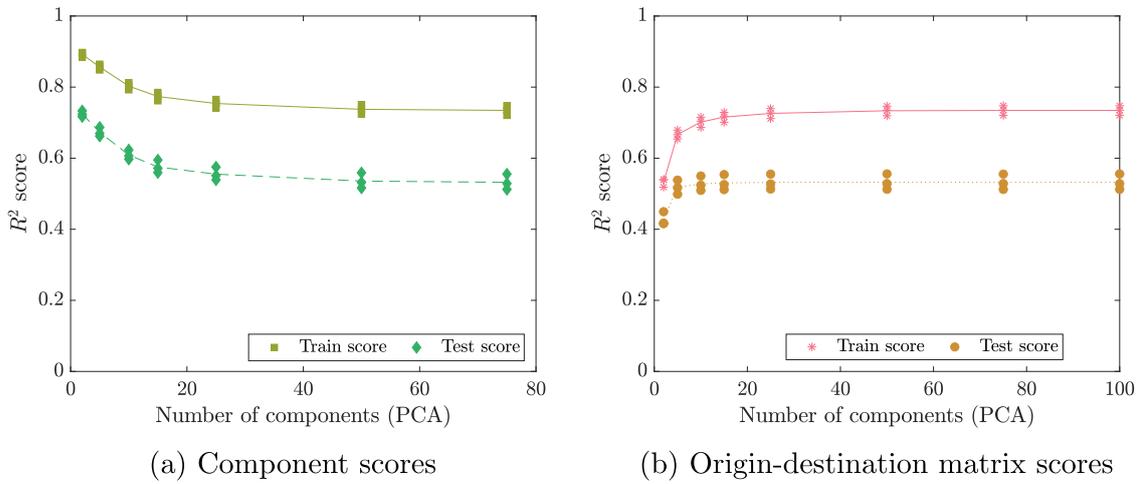


Figure 16. Performance of random forest model for the prediction of OD matrices from density heatmaps. The performance is evaluated with respect to the number of components of the PCA of the OD matrices that are predicted. The R^2 score is calculated both for the predicted components and the reconstructed entire OD matrices.

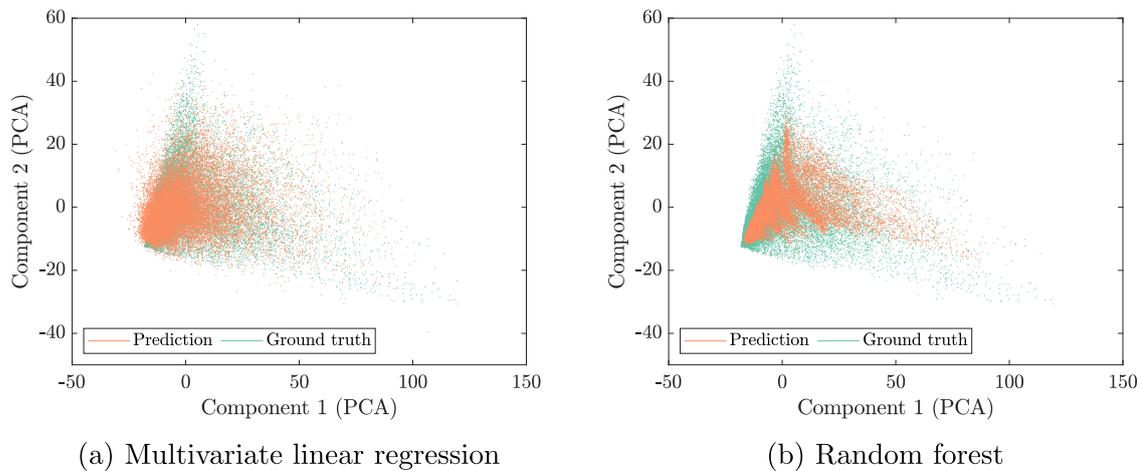


Figure 17. Scatter plot of the first and second component of the PCA of the OD matrices. The ground truth components are shown together with the prediction provided by (a) multivariate linear regression and (b) random forest.

5. Conclusion

In this work, we estimated time-dependent OD matrices from crowd density heatmaps. OD matrices describe where pedestrians come from and where they go. They serve as vital input for predictive crowd simulations. Instead of relying on full trajectories which are often not available, we constricted the input data to density heatmaps that can be derived from various sensors. In addition, the heatmaps are of constant size, which

allows the application of statistical learning models. The heatmaps were computed from real-world trajectories in a train station. We used a series of five heatmaps in a time interval of ten seconds as input for the models. Our goal was to predict the OD matrix for the same time interval. To reduce the dimensionality of the input data, we performed a PCA on the input space, maintaining an explained variance of 75% of the data.

We compared two different methods for the estimation: first, a multivariate linear regression model. Second, random forest as an example of a nonlinear model. We discovered that linear regression predicts the OD matrices with acceptable accuracy. Further investigation revealed that the linear relationship mapped in the linear regression explains only a portion of the data. We contend that a nonlinear model is required to capture the entire relationship. As a result, we used a random forest model to solve the problem, which better captured the shape of the output space but performed slightly worse in terms of our metric, the R^2 score.

It is promising to see that OD matrices can be predicted from density heatmaps which nowadays can be automatically extracted from video surveillance data. In the context of real-time crowd predictions, the estimation of OD matrices can be performed online to initialize the simulation.

The comparably low scores suggest that random forest is not the best model for this problem. In future work, we intend to use more complex machine learning models. Convolutional neural networks, which are popular for image data, and recurrent neural networks, which are tailored to time series, are potential candidates.

We discovered that the results of this study are highly dependent on the metric. In this contribution, we used a standard metric of the scikit-learn library. Future work will include developing a custom performance measure for OD matrices that reflects the effect of initialization on simulation results.

To summarize, our results demonstrate that both a linear and a non linear model can estimate some features of the OD matrices. There is no clear winner regarding the chosen metric, the R^2 score. However, the results show that some portion of the OD matrices can indeed be learned and therefore encourage further investigations including an exhaustive search for a better model. Thus, we achieve a necessary step towards real-time crowd simulations.

Acknowledgments

This research was funded by the German Federal Ministry of Education and Research under Grant Nos. 13N14464 (S2UCRE), 13N14562 (OPMOPS), and 13FH669IX6 (roVer).

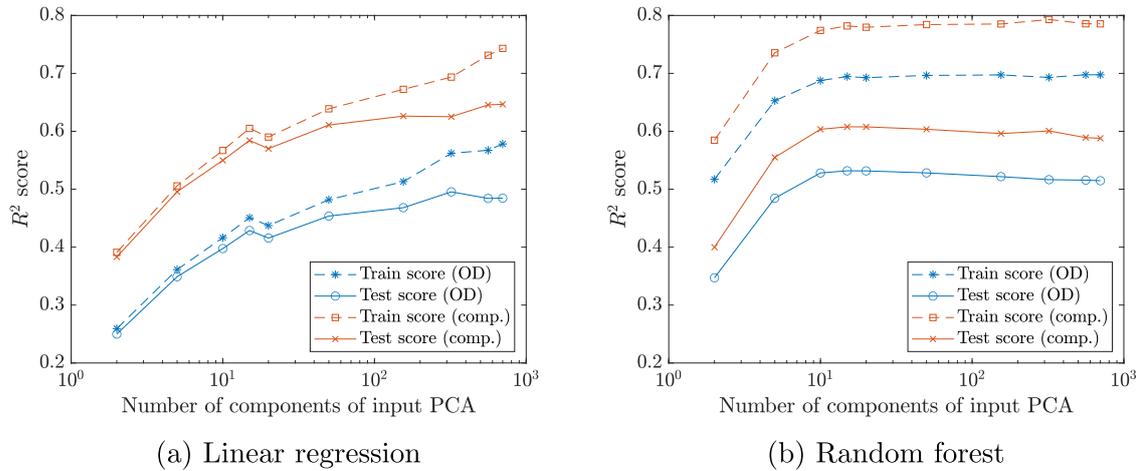


Figure A1. Performance of multivariate linear regression and random forest for predicting OD matrices with respect to the number features in the input samples. The features in the input samples are the components of the PCA decomposition of the density heatmap series. For each configuration, the R^2 scores are averaged over all three folds and the number of predicted components of the PCA of the OD matrices.

Appendix A. Impact of decomposition of density heatmap series on the model performance

As explained in section 2.4, we perform a PCA decomposition for the density heatmap series to reduce the number of features and therefore the size of the input samples. Throughout the paper, we keep so many components of the PCA decomposition so that the reconstruction reaches an explained variance of 75%. For a series of $N_t = 5$ density heatmaps, this means keeping 321 components.

We analyze the impact of the number of components kept from the input PCA on the performance of the multivariate regression and random forest in figure A1. For linear regression, the results indicate that there is a trade-off between the performance in terms of the test score and the difference between training and test score. With a larger number of components, both training and test score increase. However, the gap between training score and test score also increases. Using more than 321 components of the input PCA brings no or only negligible improvement.

Interestingly, the performance of random forest is not improved by increasing the number of components. Instead, the maximum performance (in terms of the test score) is achieved at about 15 components of the input PCA. However, the performance with 321 components is similar. Therefore, to allow a fair comparison with linear regression, we stick with 321 components.

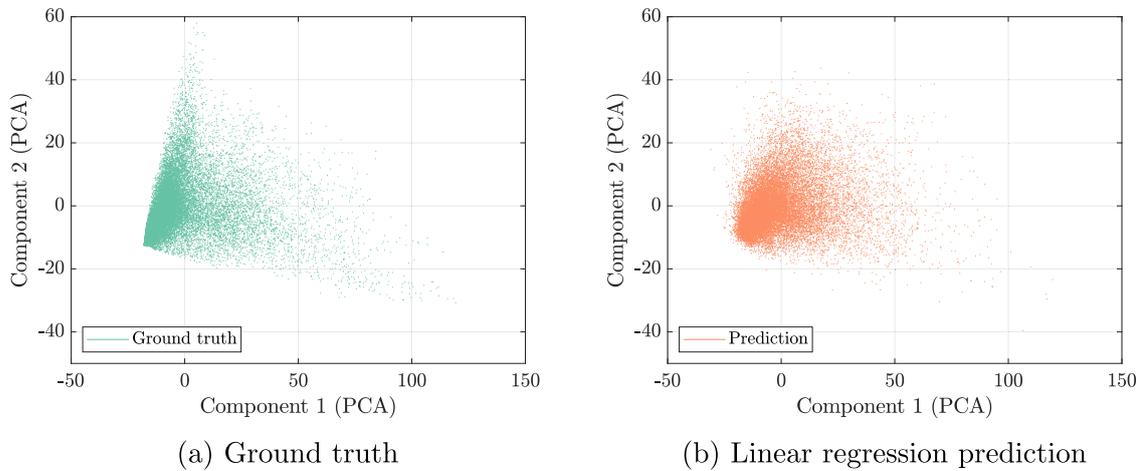


Figure B1. Scatter plot of the first and second component of the PCA of the OD matrices. The multivariate linear regression predictions of the components (a) are shown together with the ground truth components (b).

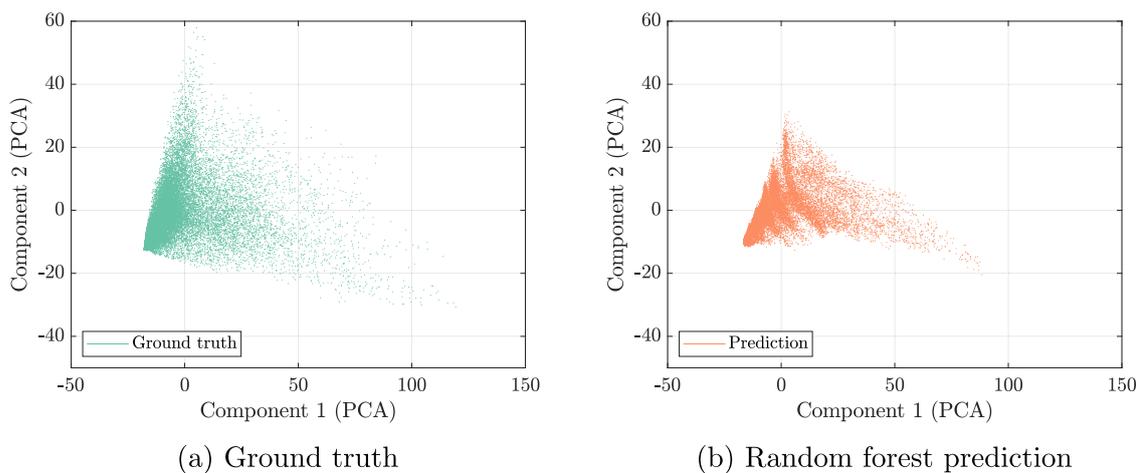
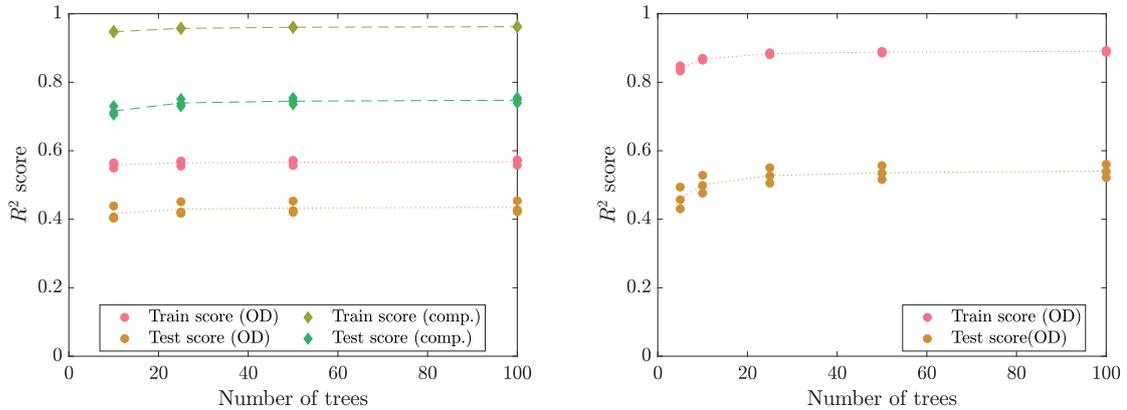


Figure B2. Scatter plot of the first and second component of the PCA of the OD matrices. The random forest predictions of the components (a) are shown together with the ground truth components (b).

Appendix B. Component analysis

We compare the a component analysis of the PCA decomposition of the OD matrices between the ground truth and the predicted matrices by linear regression and random forest. For the component analysis, we plot the first against the second PCA component. In figure B1, the predictions of multivariate linear regression are compared to the ground truth components of the OD matrices. The shapes of the point clouds differ significantly. For the nonlinear RF the two forms are similar (cf figure B2).



(a) Prediction of the first two components of the principal component analysis of the OD matrix (b) Prediction of the entire OD matrix

Figure C1. Performance of random forest for predicting OD matrices with respect to the number of trees.

Appendix C. Random forest parameters

There are three main parameters to tune the performance of random forest: first, the number of estimators or trees that are trained to the data. Second, the maximum depth for each tree. Third, the number of features is taken into account in each split. To find a suitable configuration for our problem, we investigate the impact of each parameter separately. For each parameter, we analyze the impact regarding two configurations: predicting the entire OD matrix and predicting two components of the target PCA.

C.1. Number of trees

In figure C1 the performance of random forest for different numbers of trees is shown (number of features 0.75×321 , maximum depth: 15). We choose 100 trees.

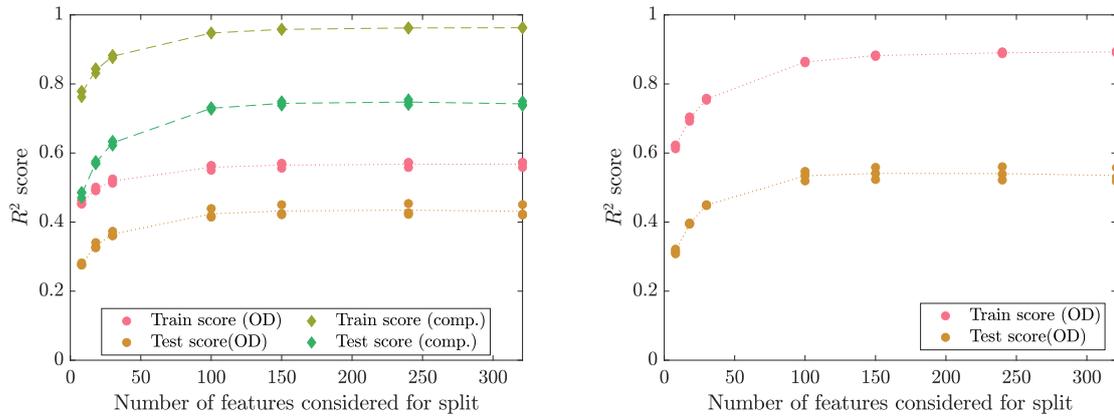
C.2. Number of features for split

Figure C2 displays the results for random forests with respect to the number of features considered when looking for the best split (number of trees: 100, maximum depth: 15). We choose all features (321) to be considered for the split.

C.3. Maximum depth of the trees

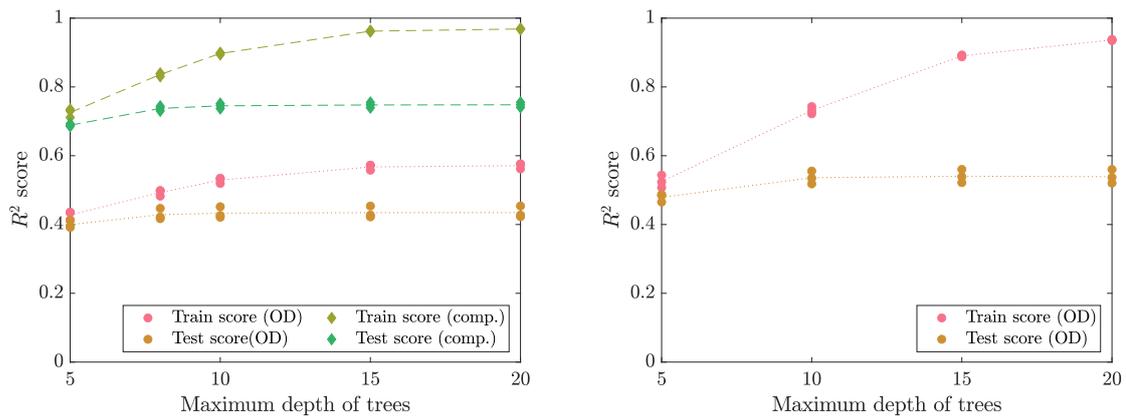
Figure C3 depicts the scores for random forest with different values for the maximum depth of the trees (number of trees: 100, max features: 0.75×321). The maximum depth has the largest impact on the results. The best test scores with a minimum distance between train and test score are obtained with a maximum depth of 10.

Toward learning dynamic origin-destination matrices from crowd density heatmaps



(a) Prediction of the first two components of the principal component analysis of the OD matrix (b) Prediction of the entire OD matrix

Figure C2. Performance of random forest for predicting OD matrices with respect to the maximum number of features considered when looking for the best split.



(a) Prediction of the first two components of the principal component analysis of the OD matrix (b) Prediction of the entire OD matrix

Figure C3. Performance of random forest for predicting OD matrices with respect to the maximum depth of the trees.

Appendix D. Overlapping data set

We generate a data set with overlapping time intervals to increase the sample size. Overlapping means that consecutive samples are no longer drawn from disjoint time intervals, but are instead drawn from time intervals of 10 s separated by one second. This mimics Takens' time-delayed embedding [32], which is a technique for reconstructing a dynamic system from a series of observations. The new dataset contains approximately 130 000 samples.

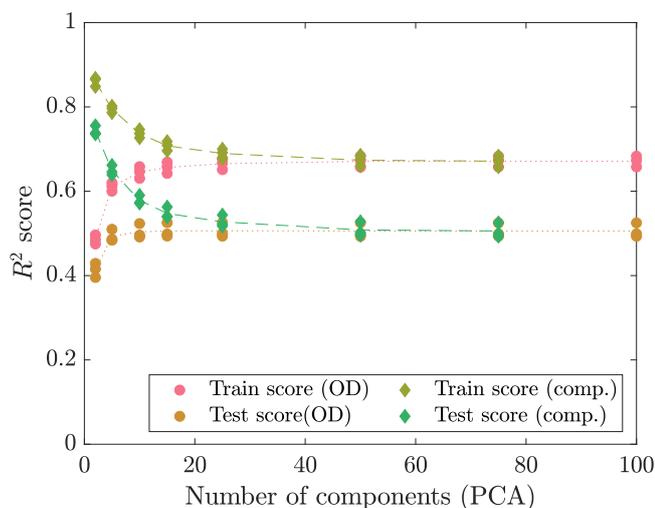


Figure D1. Performance of random forest model using overlapping time intervals for the input samples. Instead of the entire OD matrix, the components of its PCA are predicted.

Due to a large amount of input data, we use the incremental PCA approach by Ross *et al* [35] implemented in [33]. That means, the PCA is fitted iteratively with batches of $5 \times 20,520$ samples.

The results using overlapping time intervals are shown in figure D1. The performance of random forest could not be improved using overlapping time intervals.

References

- [1] Helbing D and Molnár P 1995 *Phys. Rev. E* **51** 4282–6
- [2] Hirai K and Tarui K 1975 A simulation of the behavior of a crowd in panic *Proc. 1975 Int. Conf. Cybernetics and Society* p 409
- [3] Gipps P G and Marksjö B 1985 *Math. Comput. Simul.* **27** 95–105
- [4] Seitz M J and Köster G 2012 *Phys. Rev. E* **86** 046108
- [5] von Sivers I and Köster G 2015 *Transp. Res. B* **74** 104–17
- [6] Baqui M and Löhner R 2020 Towards real-time monitoring of the hajj *Proc. 9th Int. Conf. Pedestrian and Evacuation Dynamics* pp 394–402
- [7] Kemloh Wagoum A U, Steffen B and Seyfried A 2012 Runtime optimisation approaches for a real-time evacuation assistant *Parallel Processing and Applied Mathematics* ed R Wyrzykowski, J Dongarra, K Karczewski and J Waśniewski (Berlin: Springer) pp 386–95
- [8] Kemloh Wagoum A U, Steffen B, Seyfried A and Chraibi M 2013 *Adv. Eng. Softw.* **60–61** 98–103
- [9] Pettré J, de Heras Ciechomski P, Maïm J, Yersin B, Laumond J-P and Thalmann D 2006 *Comput. Animat. Virtual Worlds* **17** 445–55
- [10] van Toll W G, Cook A F and Geraerts R 2012 *Comput. Animat. Virtual Worlds* **23** 59–69
- [11] Sindagi V A and Patel V M 2017 *Pattern Recognit. Lett.* **107** 3–16
- [12] Zönnchen B 2021 Efficient parallel algorithms for large-scale pedestrian simulation *Dissertation* Technical University of Munich (<http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:bvb:91-diss-20210521-1593965-1-9>)
- [13] Cascetta E 1984 *Transp. Res. B* **18** 289–99
- [14] Bell M G H 1983 *Transp. Sci.* **17** 198–217

- [15] Pitombeira Neto A R, de Oliveira Neto F M and Grangeiro Loureiro C F 2017 *Transportes* **25** 1344
- [16] Khan S D, Bandini S, Basalamah S and Vizzari G 2016 *Neurocomputing* **177** 543–63
- [17] Pouke M, Goncalves J, Ferreira D and Kostakos V 2016 *Comput. Environ. Urban Syst.* **57** 118–29
- [18] Li Y, Sarvi M and Khoshelham K 2019 Pedestrian origin-destination estimation in emergency scenarios 2019 *9th Int. Conf. Fire Science and Fire Protection Engineering (ICFSFPE)* pp 1–5
- [19] Chan K, Lam W H, Ouyang L and Wong S 2007 *J. E. Asia Soc. Transp. Stud.* **7** 1760–73
- [20] Wong S C and Tong C O 1998 *Transp. Res. B* **32** 35–48
- [21] Ahn Y, Kowada T, Tsukaguchi H and Vandebona U 2017 *Transp. Res. Proc.* **25** 315–30
- [22] Bauer D 2012 *IET Intell. Transp. Syst.* **6** 463–73
- [23] Hänseler F S, Molyneaux N A and Bierlaire M 2017 *Transp. Sci.* **51** 981–97
- [24] Gödel M, Köster G, Lehmborg D, Gruber M, Kneidl A and Sesser F 2020 *Collect. Dyn.* **5** 134–41
- [25] Kleinmeier B, Zönnchen B, Gödel M and Köster G 2019 *Collect. Dyn.* **4** 1–34
- [26] Boltes M, Zhang J, Seyfried A and Steffen B 2011 T-junction: experiments, trajectory collection, and analysis 2011 *IEEE Int. Conf. Computer Vision Workshops (ICCV Workshops)* pp 158–65
- [27] Gödel M, Spataro L and Köster G 2020 Can we learn where people come from? Retracing of pedestrians origins in merging situations *Technical Report* Munich University of Applied Sciences (arXiv:2012.11527)
- [28] Weidmann U 1993 *Transporttechnik der Fussgänger (Schriftenreihe des IVT)* vol 90 2nd edn (Zürich: Institut für Verkehrsplanung, Transporttechnik, Strassen- und Eisenbahnbau (IVT) ETH) (<https://doi.org/10.3929/ethz-b-000242008>)
- [29] Bosina E and Weidmann U 2017 *Physica A* **468** 1–29
- [30] Bosina E 2018 A new generic approach to the pedestrian fundamental diagram *PhD Thesis* Institut für Verkehrsplanung und Transportsysteme (IVT), ETH Zürich (<https://doi.org/10.3929/ethz-b-000296226>)
- [31] Davidich M and Köster G 2013 *PLoS One* **8** e83355
- [32] Takens F 1980 Detecting strange attractors in turbulence *Dynamical Systems and Turbulence Proc. Symp. Held at the University of Warwick 1979/80* (Lecture Notes in Mathematics) vol 898 (Berlin: Springer)
- [33] Pedregosa F *et al* 2011 *J. Mach. Learn. Res.* **12** 2825–30
- [34] Breiman L, Friedman J, Stone C J and Olshen R A 1984 *Classification and Regression Trees* (London: Chapman Hall)
- [35] Ross D A, Lim J, Lin R-S and Yang M-H 2008 *Int. J. Comput. Vis.* **77** 125–41