
Article:
Wang, H, Ondrej, J and O'Sullivan, C (2017) Trending Paths: A New Semantic-level Metric for Comparing Simulated and Real Crowd Data. IEEE Transaction on Visualization and Computer Graphics, 23 (5). pp. 1454-1464. ISSN: 1077-2626

https://doi.org/10.1109/TVCG.2016.2642963

Trending Paths: A New Semantic-level Metric for Comparing Simulated and Real Crowd Data

He Wang, Jan Ondřej and Carol O'Sullivan

Abstract—We propose a new semantic-level crowd evaluation metric in this paper. Crowd simulation has been an active and important area for several decades. However, only recently has there been an increased focus on evaluating the fidelity of the results with respect to real-world situations. The focus to date has been on analyzing the properties of low-level features such as pedestrian trajectories, or global features such as crowd densities. We propose the first approach based on finding *semantic information* represented by latent *Path Patterns* in both real and simulated data in order to analyze and compare them. Unsupervised clustering by non-parametric Bayesian inference is used to learn the patterns, which themselves provide a rich visualization of the crowd behavior. To this end, we present a new Stochastic Variational Dual Hierarchical Dirichlet Process (*SV-DHDP*) model. The fidelity of the patterns is computed with respect to a reference, thus allowing the outputs of different algorithms to be compared with each other and/or with real data accordingly. Detailed evaluations and comparisons with existing metrics show that our method is a good alternative for comparing crowd data at a different level and also works with more types of data, holds fewer assumptions and is more robust to noise.

Index Terms—Crowd Simulation, Crowd Comparison, Data-Driven, Clustering, Hierarchical Dirichlet Process, Stochastic Optimization

1 Introduction

Although a large variety of crowd simulation methods exist, choosing the best algorithm for specific scenarios or applications remains a challenge. Human behavior is very complex and no one algorithm can be a magic bullet for every situation. Furthermore, different parameter settings for any given method can give widely varying results. Subjective user studies can be useful to determine perceived realism or aesthetic qualities, but more objective methods are often needed to determine the fidelity and/or predictive power of a given simulation method with respect to real human behaviors. The hierarchical and heterogeneous nature of human crowd behaviors make it very difficult to find a definitive set of evaluation rules or empirical metrics. Therefore, data-driven evaluation methods are particularly useful for this purpose.

Previous data-driven methods tend to focus on comparisons between high-level global features such as densities and exit rates, or low-level data such as individual trajectories. In the former case, the results are often too general and do not reflect the heterogeneity of human behaviors, and in the latter case, the results are too specific to the exact scenario recorded. Based on [1], we propose a data-driven approach to crowd evaluation based on exposing the latent patterns of behavior that exist in both real and simulated data, which offers a compromise between these two extremes that takes both the global and local properties of crowd motion into account in order to facilitate a comprehensive qualitative and quantitative analysis of the data. Different from existing methods, the input of our method is not limited to trajectory data and it also holds less assumptions on the data and is more robust to noise. Finally, we provide in-depth evaluations to show that our metric is a good alternative capturing unique information which is difficult for existing approaches.

For a high-level explanation of our approach, we consider the example of a large public square with many entrances and exits, such as the train station shown in Figure 1(a). Pedestrians typically do not wander randomly, nor do they walk in straight lines; rather, they self-organize into flows or standing clusters, with each *trajectory* consisting of a series of one person's steps as he moves through the square (Figure 1(b)). A group of similar trajectories can be thought of as a *trending path* that represents the aggregation of multiple pedestrians' positions and orientations. Combining all such trending paths together will generate an overall *path pattern* that consists of flows of location-orientation pairs (Figure 1(c-g)). In scenarios where global path planning does not significantly affect behavior, e.g., walking through a corridor, local inter-personal dynamics can also lead to different path patterns. The path patterns created are therefore the result of local/internal dynamics and global/external factors.

The main contribution of our paper is a new approach to analyzing and comparing crowd data based on discovering latent path patterns. To automatically extract these patterns from both real and simulated data, we present a *SV-DHDP* model that is the first to combine a Dual Hierarchical Dirichlet Process with Stochastic Variational Inference. The patterns themselves provide a rich visualization of the crowd's behaviors and can reveal qualitative properties that would be difficult or impossible to see by simply viewing the original data. Furthermore, we propose two quantitative metrics that computes the similarity between both real and

- *H. Wang is with the University of Leeds, Leeds, UK. E-mail: h.e.wang@leeds.ac.uk*
- *J. Ondřej and C. O'Sullivan are with Trinity College Dublin, Dublin, Ireland. E-mail: Jan.Ondrej@scss.tcd.ie, Carol.OSullivan@scss.tcd.ie (corresponding author)*

The work started when all the authors were with Disney Research, Los Angeles, USA.

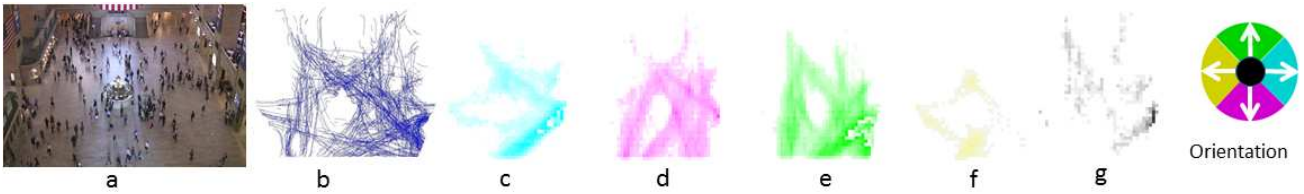


Fig. 1: (a) A video screenshot from a train station; (b) 1000 tracklets (randomly selected from 19999); (c-g) The five orientation subdomains of the top pattern as location-orientation distributions. Inset shows discretization of the orientation, with black representing zero velocity.

simulated datasets. This allows us to analyze the predictive quality of various simulation algorithms with respect to real data. We demonstrate the qualitative and quantitative capabilities of our approach on several real and simulated crowd datasets.

2 Related Work

Crowd motion properties are affected by a hierarchy of factors from geometric to cognitive [2], [3]. To model these myriad behavioral aspects, methods such as field and flow based [4], [5], force-based [6], [7], velocity and geometric optimization [8], [9], [10] and data-driven [11], [12], [13] have been proposed. Our aim is to provide an evaluation framework that imposes no assumptions on the underlying simulation mechanism and can therefore work on the output data of all such methods.

Qualitative methods for crowd evaluation have been proposed and include visual comparison [14], [15] and perceptual experiments [16], [17], [18]. Quantitative methods fall into two main categories: model-based [19], [20] and data-driven [21], [22], [23], [24]. Data-driven metrics have been proposed that use the statistics of geometric and dynamic feature analysis [25], model-based comparisons of motion randomness [26] and decision making processes [27].

Our data-driven evaluation method is partly inspired by two previous approaches. Guy et al. [26] use a dynamic system to model crowd dynamics and compute an entropy metric based on individual motion randomness distributions learned from the data. Our method differs in that we learn global path patterns from groups of trajectories, rather than individual ones. Charamlambous et.al. [28] apply a number of different criteria to detect anomalies in the data, whereas we focus on discovering mainstream latent patterns.

We also draw inspiration from the field of Computer Vision, where hierarchical Bayesian models [29], [30], [31] have been successfully employed for scene classification [32], [33], object recognition [34], human action detection [35] and video analysis [36], [37], [38]. The Hierararchial Dirichlet Process (HDP) has been successfully used in Natural Language Processing to discover candidate topics within corpora. By observing that crowd data can also be decomposed into a *bag of words*, Wang et al. [38] used a Dual HDP (DHDP) to analyze paths in video data.

There has been extensive research in computer vision and robotics on crowd analysis and we discuss some representative approaches here. Zhou et. al [39] model trajectories as linear dynamic systems and model starting positions and destinations as beliefs. The key information, belief, is manually labelled. Although the user can roughly label these areas, we suspect that a finer classification will require more extensive labelling. Furthermore, it is unclear how they such areas could be labelled in a highly unstructured space where every position on the boundary could be both a starting and an ending area. In our approach, we do not require manual labels for such beliefs. Ikeda et.al [40] models paths by first determining sub-goals and then learning transitions between sub-goals. However, their model of the crowd is solely based on the social-force model, and sub-goals are defined as points towards which many velocities converge. There may not be any such sub-goals (consider flows with no intersections), or there could be too many. Our method does not make any assumptions about the underlying behavior model or the existence of sub-goals. Other methods based on density or mean-flows [41], [42] interpret the whole field as one density map or one flow field whereas our method gives a series of weighted patterns.

3 Methodology

3.1 Model Choice

The first step towards exposing the latent path patterns in a crowd data set is to find a set of trending paths. Here, a trending path can be seen as a collection of similar trajectories. However, manually labelling clusters of trajectories would be difficult and time-consuming as we lack a good distance metric and prior knowledge of the number of patterns present. Popular unsupervised clustering algorithms, such as K-means [43] and Gaussian Mixture Models (GMMs) [44], require a pre-defined cluster number. Hierarchical Agglomerative Clustering [36] does not require a predefined cluster number, but the user must decide when to stop merging, which is similarly problematic. Spectral-based clustering methods [45] solve this problem, but require the computation of a similarity matrix whose space complexity is $O(n^2)$ on the number of trajectories. Too much memory is needed for large datasets and performance degrades quickly with increasing matrix size.

An alternative perspective is to treat a trending path as a distribution over location-orientation pairs (Figure 2). A group of trajectories connecting points A and B can be represented by a trending path modeled by Multinomial distributions over location-orientation pairs. Note in this representation, a trending path is a flow sub-field rather than a group of 2D curves. Although the trajectories are broken into individual location-orientation observations in this way, we overcome the randomness of a particular trajectory

Terms	Notation	Meaning
Agent State	w	$w = \{p, v\}$ where p and v are the position and orientation of an agent
State Space	$\mathbb{S}$	The set of all possible states. $\mathbb{S} = \{w_i\}$
Path	$\mathbb{P}$	A probability distribution over $\mathbb{S}$. $\mathbb{P}$(s)
Path Pattern	β	A mixture of paths.

TABLE 1: Terminology and Parameters

and represent such a trajectory group as one trending path. Next, we find all trending paths under the assumption that: if a trending path exists, there should be repeated location-orientation occurrences on this path. Then the problem is transformed to computing a (potentially infinite) number of Multinomial distributions. We present a non-parametric hierarchical Bayesian model that can automatically compute a desirable number of such Multinomial distributions from the data. Thus, it does not require a pre-defined cluster number and its space complexity is smaller than $O(n^2)$.

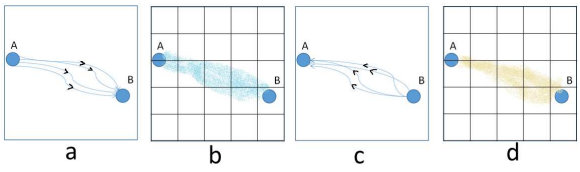


Fig. 2: Two sets of trajectories (a, c) and their corresponding trending paths modeled by Multinomials (b, d). Color coding represents different orientation sub-domains (cf. Figure 1)

We first define the terminologies in Table 1. Our SV-DHDP model employs a Dual Hierarchical Dirichlet Process, similar to that presented in [38], for pattern analysis, but we combine it with Stochastic Variational Inference (SVI) for posterior estimation that results in better performance on large datasets. In a standard hierarchical Bayesian setting, a tree is constructed in an attempt to explain a set of observations through a hierarchy of factors. In our problem, the observations are *agent states*, which we divide into equal-length *data segments* along the time domain. Our goal is to find a set of *path patterns* $\{\beta_k\}$ that, when combined with their respective weights, best describe all the segments in terms of their likelihoods. Such a tree structure is shown in Figure 3. This is a simplified figure of a three-layer Bayesian hierarchy explaining how the observations w_{d_n} can be explained by all possible patterns β_k with weights v_k. For the sake of conciseness, the full detailed version of the model is provided in the supplementary material. The overall goal here is to estimate β_ks and v_ks given w_{d_n}, which is the posterior distribution of this model p($\{\beta_k\}$, $\{v_k\}$ | w_{d_n}). We explain the posterior estimation in Section 4.

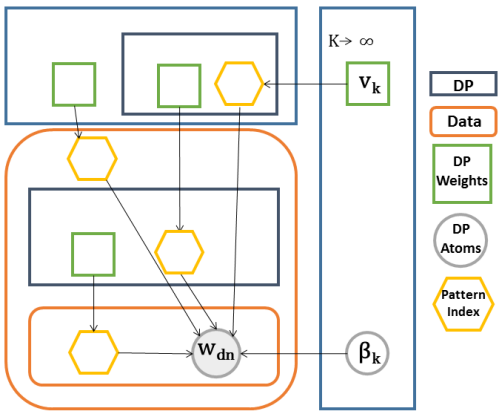


Fig. 3: SV-DHDP model. DP is Dirichlet Process. w_{dn} is the nth agent state in data segment d. K is the total number of patterns. v_k is the weight of the kth pattern. β_k is the kth pattern. Arrows indicate dependencies.

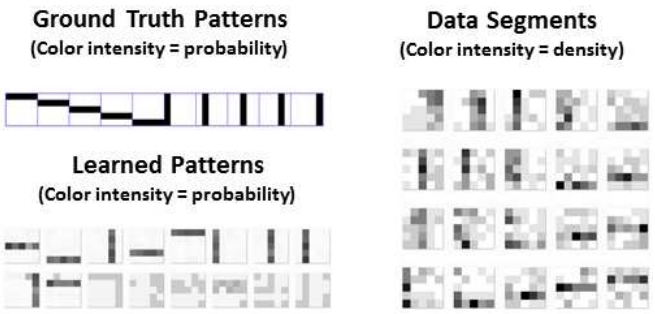


Fig. 4: Illustrative example with 100 data segments each with accumulative 50 positions: (top left) 10 ground truth path patterns; (right) example data; (bottom left) The top 16 path patterns learned

3.2 An Illustrative Example

After initial experiments using our model, we find that although many trending paths can be found in a dataset, only a subset of them are needed to describe a data segment (i.e., a time slice of the dataset). Furthermore, different subsets of the path patterns exist in different data segments. We use a simple example to illustrate this principle.

Consider again the case of a public square, simplified as a 5×5 grid environment. Imagine that there are only 10 possible paths that people will take, illustrated as horizontal and vertical bars (Figure 4 top left). Note that in this simple case each path represents a single ground truth path pattern, whereas in more complex scenes such as those presented later in this paper, a particular path usually co-occurs with different ones. For the sake of clarity, we also only cluster positions. We synthesize a dataset representing the activity in the square by randomly combining several ground truth path patterns and performing random sampling to generate 100 data segments, each consisting of 50 accumulated positions (e.g., Figure 4 right). Each data segment is a density map of positions (the darker the cell, the higher the density) and mimics an observation of the square over some time interval. We can observe the phenomenon that each segment can be described by a subset of path patterns. Applying our model, we learn all the latent path patterns from our synthetic dataset and Figure 4 (bottom left) shows the top 16 found. As we can see, the top 10 match our ground truth patterns. Although additional patterns are learned, they are less prominent (smaller intensities) and have much smaller weights, so they are ranked lower.

4 Posterior Estimation and Similarity

As discussed in Section 3.1, the novelty of our SV-DHDP model is the way we compute the posterior. There are two approaches commonly used for this purpose: sampling and variational inference. Sampling methods provide good model fitness on relatively small datasets. But the proof of convergence is still open and they have other shortcomings [46]. We therefore use a Stochastic Variational Inference (SVI) method, which is much faster on large datasets (such as crowd behaviors observed over time).

For a standard two-layer HDP model, many methods have been developed [46], [47], [48]. Our SVI technique is similar to that recently proposed by Hoffman et al. [47], except that their model is a simple two-layer HDP model whereas ours has an additional DDP layer. This extension is non-trivial and involves much more than merely adding one more DP layer to a two-layer HDP model. To our knowledge, this is the first attempt to apply variational inference on this type of model. Please refer to the supplementary materials for detailed mathematical derivations and algorithms.

4.1 Model Fitness

By dividing a dataset into training data C_{train} and a test data segment C_{test}, we can evaluate the model fitness by the predictive likelihood of C_{test}. We further divide C_{test} into two sets of samples: observed w_i^{obs}, and held-out w_i^{ho}. We also keep the unique state sets of the two sets disjoint. We first use C_{train} to train our model to compute the approximate posterior, and then use w_i^{obs} and the approximate posterior to fine tune the top-level path pattern weights. Finally, we compute the log likelihood of w_i^{ho}. This metric gives a good predictive distribution and avoids comparing parameter bounds. Similar metrics are used in [46], [47], [48] for evaluating model fitness. It is computed by:

$$\begin{aligned} & p(w_j^{ho}|C_{train}, w_j^{obs}) \\ &= \int\int (\sum_{k=1}^{K} v_k \beta_{k,w_j^{ho}}) p(v|C_{train}, \beta) p(\beta|C_{train}) dv d\beta \\ &\approx \int\int (\sum_{k=1}^{K} v_k \beta_{k,w_j^{ho}}) q(v) q(\beta) dv d\beta \\ &= \sum_{k=1}^{K} \mathbb{E}_q[v_k] \mathbb{E}_q[\beta_{k,w_j^{ho}}] \end{aligned} \tag{1}$$

where K is the truncation number at the top level, $\beta_{k,w_j^{ho}}$ is the probability of the state w_j^{ho} in path pattern β_k, and q is the variational distribution. For a testing data segment, per-state log likelihood $\prod_j p(w_j^{ho}|C_{train}, w_j^{obs})$ is computed. When training the model, we plot per-state log likelihood and stop the optimization when it becomes stable.

4.2 Inference Based Similarity Metric

In addition to extracting path patterns, we would also like to propose a metric for measuring similarities between datasets, so that the quantitative similarity of simulation vs. simulation, real data vs. simulation. or even real data vs. real data can be computed. Since our model can compute path patterns for two datasets, a naive approach is to use some commonly used metric such as KL-divergence or even plain Euclidean distance between pairs of patterns. However, we can easily end up with two sets of different sizes; and comparing two sets of probabilistic distributions is not a well-defined problem. Furthermore, while it may seem like a good idea to only compare the top n patterns from both pattern sets, this would be unfair because the patterns are weighted differently within their sets and the choice of n is unclear. A more elegant metric is therefore needed to compare two datasets.

We know that to evaluate model fitness on dataset A, we would use a test data segment from A. This model fitness also implies that if dataset B has similar path patterns to A, then the data from B should also give a good likelihood in Equation 1. In this way, we can compute the per-state predictive likelihood of B given A:

$$lik(B|A) = p(w^{ho}|A, w^{obs}). \tag{2}$$

Here we replace C_{train} in Equation 1 with A. Both the observed data w^{obs} and the held-out data w^{ho} are from B instead of A. This metric resolves the two concerns mentioned above. In addition, since our patterns are Multinomials, it is always possible to do pair wise comparison such as KL-divergence and Root Mean Squared Error if needed.

5 Path Pattern Abstraction

To show the generality and robustness of our method, we apply it to both simulated datasets as well as real data from various scenarios with different features on different noise levels. We also compare our methods with existing approaches and discuss performance. All our patterns are color-coded in the various figures, with different colors represent orientation as in Figure 1 where color intensities show probabilities.

5.1 Simulation Datasets

Real data exhibits both global and local features, caused by the fact that pedestrians tend to plan their paths through an environment based on external factors such as entrances, exits and personal goals, but they are often deflected from their paths due to the necessity to avoid members of a crowd. In simulations, different types of simulation algorithms are used to model local steering behaviors and global path planning strategies. We explore the effects of these algorithms separately by first varying local steering methods while minimizing the impact of the global path planning. Then we fix the steering behavior and vary the global path planning strategies.

5.1.1 Local Steering

We choose four steering algorithms that are representative of commonly used methods: MOU09 [49] is a recent version of Helbing's social force model; PETT09 [9] is a velocity obstacle method based approach similar to RVO; ONDREJ10 [10] uses bearing angle to avoid collisions; and PARIS07 [50] solves steering in velocity space. Many other methods exist, such as potential fields, fluid based, hybrids, foot-step

planning, but our goal is not to analyze every possible approach, but to demonstrate how our method can capture the differences produced by different reactive steering methods.

We set up a bi-directional flow experiment to show our analysis for local steering behaviors. Two rectangular areas are placed at the top and bottom of a scene (Figure 5) and two groups of agents are created. For one group, agents are randomly generated in one area with randomly selected destinations in the other area, thus avoiding any complex global path planning. For the other agent group, we switch the generation and destination areas. This forces both agent groups to use steering behaviors in order to avoid the others. Each simulation lasts for around 25 minutes and involves 20000 agents.

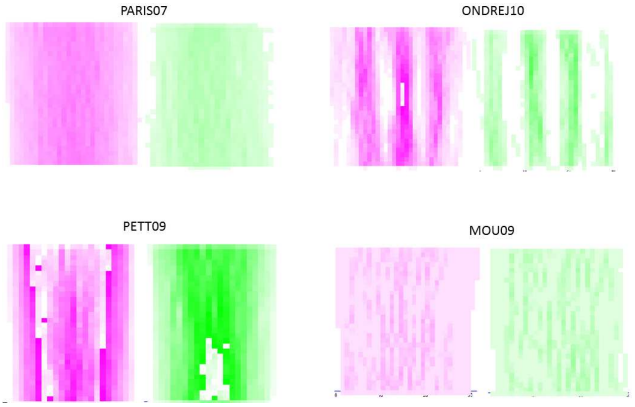


Fig. 5: Top path patterns from the data created by four representative local steering algorithms

Snapshots of the simulation data can be found in the supplemental material. Figure 5 shows the top path patterns computed. Intuitively, we can see that PARIS07 does not give prominent patterns meaning the crowd is spreading out all the time. ONDREJ10 tends to give stable flows compared to other methods. And PETT09 and MOU09 are in the middle because their patterns are slightly more concentrated than PARIS07, but less so than ONDREJ10. PARIS07 looks more similar to PETT09 and MOU09 than ONDREJ10. This visualization thus facilitates a qualitative understanding of the behaviours generated using different local steering mechanisms. Later we will see how we can also quantitatively compare them with each other.

5.1.2 Global Path Planning

In this experiment, we fix our local steering model [51] and vary the global path planning methods to test our analysis model. The environment is a square with several obstacles in the middle. We set up a generation area at the top and destination area at the bottom. Also, we recycle 64 agents over and over again to generate 400 second data. Three global path planning methods, Navigation mesh [52], Roadmap [53] and Potential Field [54] are used here, referred as Navmesh, Roadmap and PoField. Trajectories are generated using Menge [55] (Figure 6) and the top patterns found are shown in (Figure 7).

The top patterns of all three methods are down-going flows as expected but they spread out within the environment in slightly different ways. In addition to these high

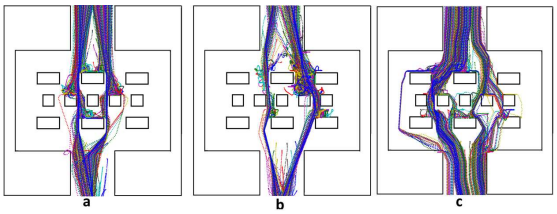


Fig. 6: Trajectories created by three global path planning algorithms: Navmesh, Roadmap and PoField

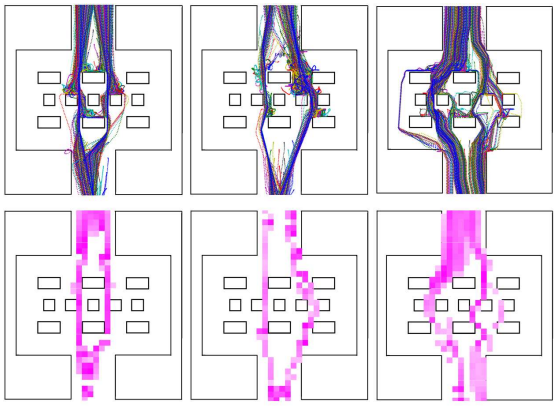


Fig. 7: Top path patterns from the data created by three global path planning algorithms: Navmesh, Roadmap and PoField

probability patterns, other patterns are also learned and we do find other colors, albeit with much smaller weights. These patterns occur when agents get completely blocked so they start to walk in other directions to find their way out.

5.2 Real Trajectory Data

In addition to simulation data, we also show experimental results computed on two real datasets. Although trajectories can be estimated from many different sensors such as videos, GPS, etc., we use videos here because cameras are widely available. The first dataset is a 6 minute video clip of 967 pedestrians in a park recorded by a mid-distance camera in a park. We manually annotate the trajectories so that we have relatively complete trajectories with very little noise.

The second dataset consists of 19999 tracklets recorded in New York Grand Central Terminal by a far distance camera [42] (downloaded from http://www.ee.cuhk.edu.hk/ xgwang/grandcentral.html). The trajectories are computed based on moving pixels and contain only partial and noisy tracklets, thus demonstrating the robustness of our method.

5.2.1 Park

All trajectories and some data segments are shown in Figure 8. To train the model, the truncation numbers for J, I, L and K (parameters explained in the supplementary material) are set to 10, 15, 4 and 20 respectively. The training took 0.58 hours and Figure 9 Reference shows some high-weight patterns. There are several major flows learned from the data. One is the flow going from 3 to 2 (References b and j). They mainly differ in whether they go through the

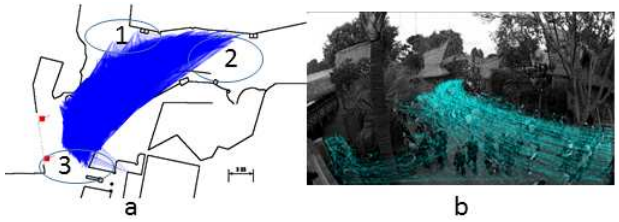


Fig. 8: Park dataset: (a) Projected trajectories, (b) Annotated trajectories overlaid on a frame from the video. Red dots are cameras.

narrow corridor along the bottom or not. Another up-going flow is Reference f from 3 to 1. The major down-going flows are from 2 to 3 (magenta and yellow). These paths are also observed in the data.

5.2.2 *Train Station*

The whole area is a square with each dimension approximately 50m long. We discretize the domain into 1×1 meter grids and set J, I, L and K (parameters explained in the supplementary material) to 10, 15, 3 and 20. The training took 1.83 hours. Some patterns are shown in Figure 1.

In Figure 1, **e** is the major up-going flow and **d** is the major down-going flow. Both are observed in the data. The right-going flow shown in Figure 1 **c** is another major flow observed in the data. Interestingly, the left-going flow pattern (yellow) is not very prominent. After looking at the data, we found that since it shares boundaries with green and magenta, some of the left-going flows are captured in Figure 1 **d** and **e** instead.

5.3 Optical Flow Data

Trajectory computation from video data is sometimes not reliable due to different lighting conditions and occlusions which makes it difficult to get the data for path pattern recognition. In such cases, optimal flows can be automatically computed and used where the path patterns become optical flow patterns. A good feature of optical flows is that different from estimating trajectories it allows a greater flexibility of the position and orientation of camera. For instance, when the camera has a bird's eye view (the angle between the camera normal and the ground normal is nearly 180 degrees), the optical flow data can be very similar to trajectory data;when it is more tilted (the angle between the camera normal and the ground normal is far less than 180 degrees), it still can provide good information about the crowd flows. Some snap shots of the optical flow raw data from our Park dataset are shown in Figure 10.

In this experiment, the domain is the whole image. We discretize the image into a 10×10 pixel grid. The choice of resolution is a trade-off between modeling granularity and training performance. It also depends on the camera position and orientation. Smaller grids capture more detailed motion information but on the other hand slow down the training by generating more data and sometimes start to lose meanings when even a single motion occupies a relatively large area on the screen. We set J, I, L and K to 10, 15, 3, 20 respectively and run the experiment for 1.37 hours.

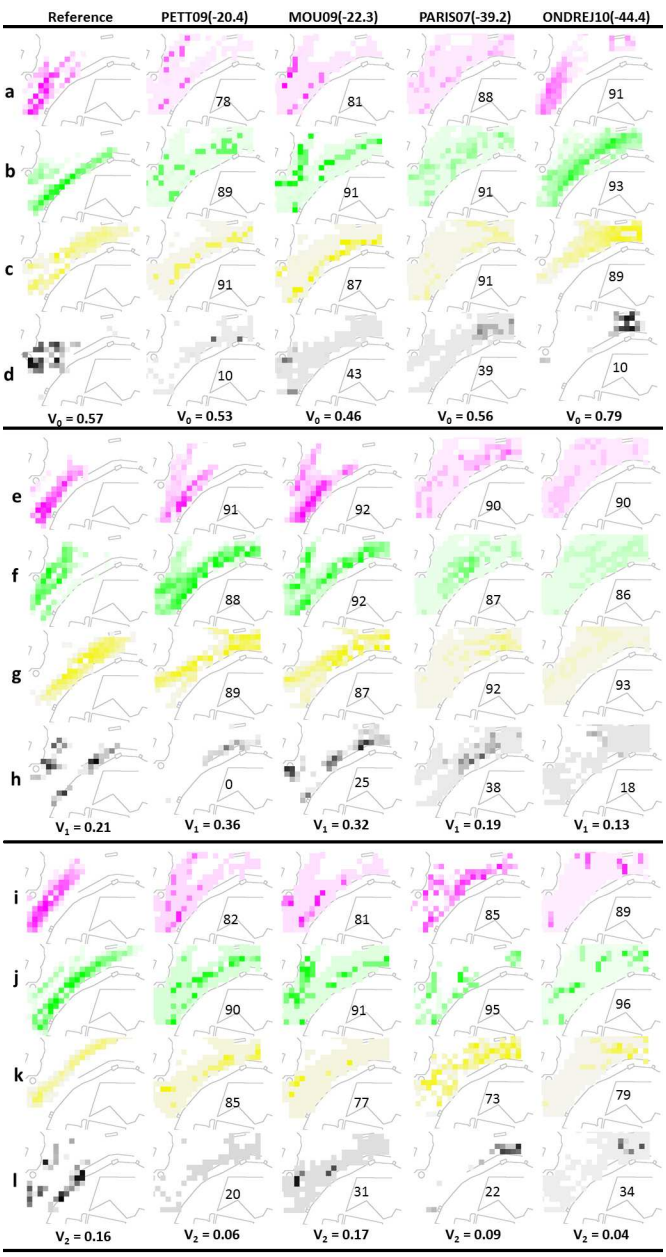


Fig. 9: Top 3 patterns for the Park dataset that cover more than 90% of weights. Each pattern is shown for 4 directions in a group (4 rows – Blue is omitted because no significant pattern found for that direction). Column 1: Top patterns from real dataset. Columns 2-5: Top patterns from four simulated datasets. Similarity scores using the real data as reference are shown in the brackets next to the name of the method. They are log likelihoods. The larger (closer to 0) the better. At the bottom of each group, weights for corresponding patterns are given. The percentages are computed by KL-divergence between a reference pattern and a simulation pattern, then normalized to 0-100. Intensity represent probability. The higher the intensity, the higher the probability.

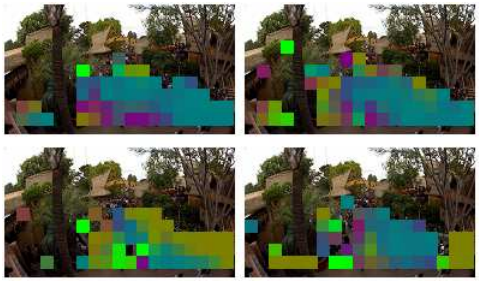


Fig. 10: Data segments of optical flows in image grids. Color coding also indicates different orientations, but in the image space.

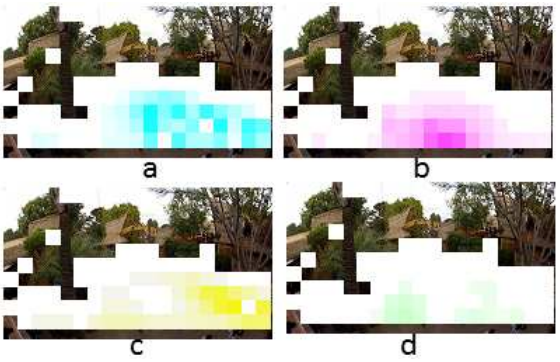


Fig. 11: The top pattern from SVI in different orientation domains. Pedestrian regions are masked using while grids.

The top four patterns are shown in Figure 11. According to the data, there is one major down-going flow and spreading out near the exit at the bottom. This is shown by Figure 11 **a** and **b**. A major up-going flow is mainly on the right of the screen which is shown by the high probability areas in Figure 11 **c**. Interestingly, given the camera position and orientation shown by the red squares in Figure 8 Left, we can see that both the down-going and up-going flows are also confirmed by our patterns computed out of trajectories Figure 11 (a) and (b). A slightly less prominent up-going flow is also captured in Figure 11 **d**. Note that there is no obvious paths in the optical flow patterns as opposed to trajectories due to the position and the orientation of the camera. However, when trajectories are not available, it is also an alternative way to analyze crowds and evaluate simulations because optical flows can be easily computed.

5.4 Comparison with previous approaches

We empirically compare our SVI method with Gibbs sampling in [38] on the train station dataset. Due to the nature and stochasticity of these two methods, it is hard to compare them in one standard setting. So we use slightly different settings for the two experiments. For the first one, we run both of them until they give similar results and compare the time cost. For the second one, we run them for roughly the same period of time and compare the results.

For the park dataset with optical flows. We run sampling until meaningful patterns appear. Since the data is dense, it takes 12.4 hours to run 10 iterations. In Figure 12, the sampling method learns similar patterns as in Figure 11,

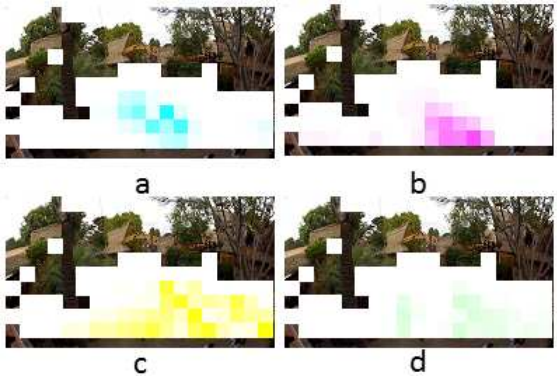


Fig. 12: Top Pattern from sampling in different orientation domains.

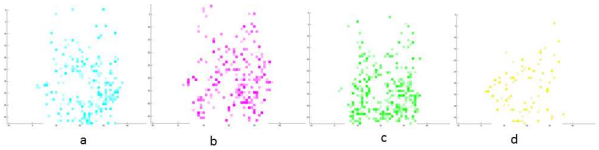


Fig. 13: The top pattern in different velocity domains learned by sampling. More patterns are available in the supplementary material.

especially Figure 12 **b** and Figure 11 **b** (both concentrated at the bottom) and Figure 12 **d** and Figure 11 **d** (both in general split into two parts). We also find patterns Figure 11 **a** and Figure 11 **c** in the data. However, Figure 12 **a** is not as spreading-out as Figure 11 **a** and Figure 12 **c** is not as concentrated on the right as Figure 11 **c** is.

For the train station, we first run the sampling for 1.95 hours and show the results in Figure 13. We made our best effort to find informative and similar patterns in the results. Compared to the patterns in Figure 1, we cannot find any pattern that are as informative. Another interesting difference is that patterns shown in Figure 13 are in general more concentrated into individual grids (reflected by their intensities compared to the ones in Figure 1), and do not fully cover the areas of the paths. We believe this is due to every state sample being given only one pattern label in sampling while in SVI each state sample has a distribution over all patterns. Also, after 1.95 hours, a total number of 198 patterns are learned and the number continues to go up to 735 after running for 40 hours, clearly showing it is not converged yet. More patterns are available in the supplementary material.

We also compared the performance of SVI with sampling. SVI is faster mainly because in every iteration, it uses a batch number that is usually much smaller than the number of data segments. In contrast, sampling uses all of them. Figure 14 shows how quickly our SV-DHDP converges. We show plots for two examples. For both synthetic and real data, our model converges at between 20 to 60 iterations.

6 Similarity Analysis

6.1 Similarity Metrics

Our similarity metric is used to provide meaningful comparisons between real reference data and simulation data.

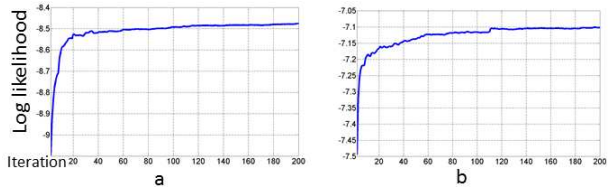


Fig. 14: Model fitness plot with iterations. a: Train Station. b: ONDREJ10 in Section 5.1.1.

compWe used the four models in Section 5.1.1 in combination with one global path planning method [56] to simulate the crowds in the park and the train station. We modelled the environments by observing the videos carefully, then randomly generating agents within the entrance areas and randomly selecting destinations within the exit areas. All similarities are computed using the real dataset as the reference. Snapshots of data segments for both experiments are shown in the supplementary material. The similarity scores for the park and train station simulations are shown in Figure 9 and Figure 15. Some top patterns are shown in Figure 9 column 2-5 and Figure 15 column 2-5.

First we emphasize that the similarities presented here are not designed to provide any kind of conclusive statement of which simulator is the best. Path patterns are affected by many factors and we did not exhaustively try all combinations of all parameter settings. For instance, it is difficult to accurately calibrate parameters including accurate entrances and exits, timing of arrival, the proportions of population in different flows and so on. After first looking at the computed patterns and scores we adjusted the entrances and exits more carefully to ensure the best performance possible for all algorithms and we speculate that the simulations could be even more improved by adjusting timing and population density. This also demonstrated how our metric can help to design simulations, because we can identify the key elements to adjust by looking at the visual patterns.

To make good use of our metric for simulation, we suggest two ways to interpret the patterns, by using Equation 2 and by computing KL-divergence between pairs of patterns to help in interpreting the visual data.

Equation 2 shows the average likelihood of the testing data. There are several major factors affecting the score. Firstly, the global path planning has a great influence. One example is Figure 15 (a):Reference where a wide flow can be seen going from the bottom to the right. In the simulation, only PETT09 roughly captures it which contributes to its score. In addition, the relative numbers of agents on each path pattern also influences the similarity. Figure 15 (a):PARIS07 has several flows that are not seen in the real data pattern. After watching the video, we found that there are only a few people walking on these paths but the simulation assigned a large number of agents to them, thus contributing to the low score of PARIS07. Next, in Figure 15, all simulations other than PARIS07 tend to form narrower paths than the real data, whereas in the park simulation, some of them are wider than the real data such as Figure 9 (b):ONDREJ10. Some of them are about the same width such as Figure 9 (f):MOU09 and some of them are too narrow such as Figure 9 (b):MOU09. The path width is affected by

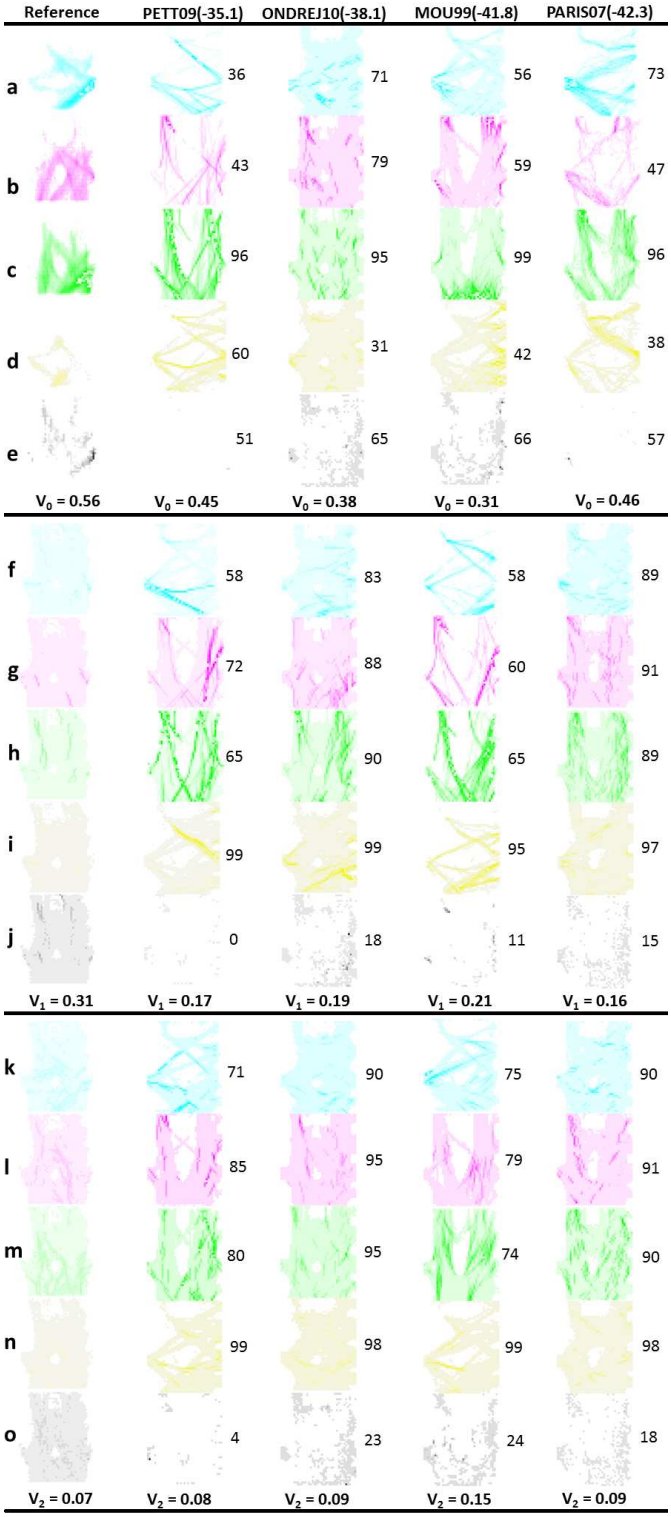


Fig. 15: Train Station patterns for real and simulated datasets. The layout and scores are computed in the same way as in Figure 9.

	PARIS07	ONDREJ10	PETT09	MOU09
SVDHDP	-39.2	-44.4	-20.4	-22.3
ENT	0.6	0.78	0.76	0.75
ENERGY	0.79	0.7	0.9	0.61

TABLE 2: Scores on PARIS07, ONDREJ10, PETT09 and MOU09 by SVHDP (our method), ENT [26] and ENERGY [57]. SVDHDP and ENERGY: the bigger the better. ENT: the smaller the better.

the simulation method itself as well as the number of agents on that path too. Finally, when it comes down to a single path, some models tend to form prominent patterns more than others, as seen in the bi-directional flow example. This also contributes to the scores. In addition, the weights are used for two purposes: analysis and comparison. Within a single dataset, the weights reflect the relative likelihoods of each path pattern. For instance, the likelihood of observing an agent on Figure 9(a):Reference is more than twice as that on Figure 9(e):Reference, indicated by their respective weights $v_0 = 0.57$ and $v_1 = 0.21$. For comparison, the weights are also considered by Equation 2.

Aside from Equation 2, the user might want simply focus on some pattern similarities. This can be computed by KL-divergence between pairs of patterns. In both Figure 9 and Figure 15, each pattern is given a score comparing itself with the corresponding pattern in the reference. We normalize the values to 0-100, where bigger is better. The results of this metric may sometimes seem contradictory with the previous one because the focus is different. For instance, in Figure 9, ONDREJ10 has the lowest similarity score. But for KL-divergence similarity, its first three patterns outperforms other datasets. This means we were able to reproduce some major flows faithfully in the reference data by ONDREJ10, but it does not do well on capturing the other sub-dominant flows. However, if the user just wants to reproduce the major flows, then ONDREJ10 is going to be a good option in this case. PETT09 and MOU09 also capture good flows in the second group. So they might be the choice if those flows are to be reproduced. For the KL-divergence metric, the weights are less meaningful because it can be applied on any pair of patterns from different datasets depending on the application.

Overall, the two metrics here focus on different aspects of the data. The similarity score gives overall performance, which is the per-state likelihood. The KL-divergence similarity emphasizes more on visual similarities. Together, they provide enriched information for different use cases.

6.2 Comparison With Existing Metrics

We also compare our similarity score with existing metrics [26] and [57]. Kapadia et al. [57] suggested a metric that compares the energy expenditure of simulated crowds and the optimal solution. Although it is not designed to directly compare simulated and real data, it does provide the possibility of placing the energy expenditure of both onto the same spectrum to compare. Guy et al. [26] proposed an entropy-based metric which is aiming for comparing simulated and real data.

We compute scores using the park dataset and they are shown in Table 2. For simulator settings, we use a standard set of parameters across different simulators. For simulator-specific parameters, we set them based on heuristics to reach best performances. Next we compare the three metrics from level of abstraction, robustness against noise, interpretability and performances.

Level of Abstraction As mentioned before, our method is based on the path information rather than individual trajectories. It is higher level information which overcomes the individual motion randomness while the other two are based on individual trajectories. Note although ENT considers randomness, it is averaged per-agent simulation error which is still based on individual trajectories.

Robustness against Noise Both ENT and ENERGY require complete trajectories that are hard to find in datasets like the train station. Moreover, a lot of publicly available datasets have only estimated trajectories from videos which suffer from occlusions and tracking errors. It is hard to compute meaningful scores using ENT and ENERGY on this kind of datasets because some of their assumptions can be easily broken. For instance, they assume the first point of a trajectory is where an agent starts and the last point is the destination. This assumption cannot hold when tracklets are present especially when there is no good way to stitch them into trajectories. Our method is also influenced but far less by the noise because we only need individual observations for recognizing the patterns.

Interpretilbility For ENERGY, we also compute the score of the real data which is 0.65. Since this metric is designed to reflect the energy efficiency, it is not very intuitive to interpret the similarities here. The real data has the second worst score because in the real situation people could not keep walking on the optimal paths. Even if we use the score of the real data as the baseline and compute the scores of simulators with respect to it, the results are still hard to interpret because a smaller energy difference does not necessarily mean a more similar crowd motion in this case. ENT provides more intuitive scores since it gives the best score if the simulator can exactly replicate the real trajectories. However, it is not desirable for situations where less or more agents are needed for simulations. In addition, since the trajectories in the park dataset contains a lot of almost-straight lines, all the simulators perform similarly, which means ENT might not be a good metric for this kind of situations. SVDHDP does not require the simulator to achieve the optimal paths or exactly replicate the trajectories or having the same number of agents. As long as flows are simulated, SVDHDP gives high scores.

Performance ENT cannot handle large datasets as fast as we can because theoretically the Expectation-Maximization (EM) they used is similar to full variational inference which is proven to be much slower than stochastic variational inference, especially when the size of the data goes up. For this reason, we used 1/3 of the park data (296 trajectories) for ENT which still took on average 0.94 hours for each simulator. Also, SVDHDP just has to train the model once on the real data. The score then can be computed in seconds while ENT needs to do EM for each simulator which makes it more computationally expensive.

7 Limitations

Our method does have several limitations. First, our metric is more affected by the environment than the metrics that only measure the intrinsic properties of crowds and ignore the environment. The main reason is our path patterns are not independent of the environment. However, we argue that the real data are almost always affected by the environment in real-world applications. In addition, when only local flow patterns are needed, our metric still works well as shown in the bidirectional flow example. Although a change of simulation setting (e.g., a global shift of flows) can result in lower scores, which does not happen to some other metrics, we argue that they can always be aligned if a rotation/translation invariant comparison is needed.

Our method does not directly measure individual trajectories thus does not reflect individual visual similarities. However, we argue that it is by design because to measure individual similarity, one has to deal with factors such as the individual motion randomness and data noise. We intend to overcome these difficulties by abstracting the information at a higher level.

Our model does not capture temporal information such as changes of patterns over time. In real applications, sometimes a path at time t means there is a high probability of it appearing again in future, the *Markovian* property of paths, which is not captured in our current model. Also, a certain path pattern can appear/disappear several times during observation. This kind of global temporal patterns are also not captured in our model.

Lastly, our truncation-based stochastic variational inference can be sometimes sensitive to the initialization even if the stochasticity in the gradient estimation helps to some extent. In practice, we did grid search to find out good initializations.

8 Conclusions and Future Work

In summary, we propose path patterns as a new perspective for comparing simulated and real crowd data, which is an abstraction of information at a level that is higher than individual trajectories but lower than global properties such as density. This compromise captures both local and global features of crowd motions thus a good metric for comparison. To tackle the challenge of path pattern computation, we propose a non-parametric hierarchical Bayesian model to automatically extract a desirable set of patterns, which themselves are informative visualizations of crowd motion patterns. To deal with large datasets, we propose a stochastic variational inference method. Besides, we propose two similarity metrics for comparison that handle the overall similarity and individual path pattern similarities for different application scenarios in crowd simulation. Finally, we did detailed comparison with existing metrics to show that our method works on more types of data, has less assumptions on the data and more robust to data noises.

In terms of data types, although our method works on both trajectories and optical flows, we do not believe one is necessarily better than the other. Optical flows can be automatically computed and impose less restrictions on the cameras. But it makes the data much denser than needed and thus slows down the training. Also, optical flows are calculated based on object movements, not only human movements, thus could contain large noises. On the other hand, trajectories can be computed or annotated that might suffer less from inaccuracy and noise depending on the algorithm and data acquisition. Although annotation can be slow and laborious, they have high qualities and makes it easier for our model to capture the path patterns quickly. Another advantage is more information can be appended onto annotated trajectories, such as interactions and activities, which gives a new possibility of clustering trajectories with activities together.

One future direction will be an extension of the current model into a dynamic model. Currently, all data are considered at once. But in real situations, the path patterns and their respective weights can change over time. To capture this effect, a dynamic model is needed. Another direction is introducing pattern merge and delete during optimization to find better solutions. To use our metric to guide simulation more automatically, we could use patterns as guiding flows for crowd simulation to improve the scores by methods such as [58]. Sampling location-orientation pairs from the learned latent patterns should be straightforward. Since the samples are more likely to be in the high probability areas, they can be used as intermediate goals to force the simulation to form flows. However, it would require more research if whole trajectories are to be sampled. In addition, we only capture flows although individual trajectories may also influence perceptual realism. A perceptual study is needed to investigate under what circumstances a flow's influence becomes dominant. A good direction is to try to capture information on both levels. Finally, we would like to add social activity and environmental information such as talking and pouring a cup of coffee to incorporate behavior patterns in the model. We believe it will further help in simulating realistic crowds with diverse behaviors.

References

[1] H. Wang, J. Ondřej, and C. O'Sullivan, "Path Patterns: Analyzing and Comparing Real and Simulated Crowds," in *I3D*, 2016, pp. 49–57.

[2] M. Kapadia, N. Pelechano, and J. Allbeck, *Virtual Crowds: Steps Toward Behavioral Realism*. Morgan & Claypool Publishers, Nov. 2015.

[3] J. Funge, X. Tu, and D. Terzopoulos, "Cognitive Modeling: Knowledge, Reasoning and Planning for Intelligent Characters," in *SIGGRAPH*, 1999, pp. 29–38.

[4] R. Narain, A. Golas, S. Curtis, and M. C. Lin, "Aggregate Dynamics for Dense Crowd Simulation," *ACM Trans. Graph.*, vol. 28, no. 5, pp. 122:1–122:8, 2009.

[5] A. Treuille, S. Cooper, and Z. Popovi, "Continuum Crowds," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 1160–1168, 2006.

[6] D. Helbing and P. Molnár, "Social Force Model for Pedestrian Dynamics," *Phys. Rev. E*, vol. 51, no. 5, pp. 4282–4286, 1995.

[7] I. Karamouzas, P. Heil, P. v. Beek, and M. H. Overmars, "A Predictive Collision Avoidance Model for Pedestrian Simulation," in *Motion in Games*, 2009, pp. 41–52.

[8] J. van den Berg, M. Lin, and D. Manocha, "Reciprocal Velocity Obstacles for real-time multi-agent navigation," in *IEEE ICRA*, 2008, pp. 1928–1935.

[9] J. Pettré, J. Ondřej, A.-H. Olivier, A. Cretual, and S. Donikian, "Experiment-based Modeling, Simulation and Validation of Interactions Between Virtual Walkers," in *SCA*, 2009, pp. 189–198.

[10] J. Ondřej, J. Pettré, A.-H. Olivier, and S. Donikian, "A Synthetic-vision Based Steering Approach for Crowd Simulation," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 123:1–123:9, 2010.

[11] K. H. Lee, M. G. Choi, Q. Hong, and J. Lee, "Group Behavior from Video: A Data-driven Approach to Crowd Simulation," in *SCA*, 2007, pp. 109–118.

[12] A. Lerner, E. Fitusi, Y. Chrysanthou, and D. Cohen-Or, "Fitting Behaviors to Pedestrian Simulations," in *SCA*, 2009, pp. 199–208.

[13] S. Kim, A. Bera, A. Best, R. Chabra, and D. Manocha, "Interactive and Adaptive Data-driven Crowd Simulation," in *2016 IEEE VR*, March 2016, pp. 29–38.

[14] S. Kim, S. J. Guy, and D. Manocha, "Velocity-based Modeling of Physical Interactions in Multi-agent Simulations," in *SCA*, 2013, pp. 125–133.

[15] S. Lemercier, A. Jelic, R. Kulpa, J. Hua, J. Fehrenbach, P. Degond, C. Appert-Rolland, S. Donikian, and J. Pettré, "Realistic Following Behaviors for Crowd Simulation," *Comp. Graph. Forum*, vol. 31, no. 2, pp. 489–498, 2012.

[16] R. McDonnell, M. Larkin, S. Dobbyn, S. Collins, and C. O'Sullivan, "Clone Attack! Perception of Crowd Variety," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 26:1–26:8, 2008.

[17] S. J. Guy, S. Kim, M. C. Lin, and D. Manocha, "Simulating Heterogeneous Crowd Behaviors Using Personality Trait Theory," in *SCA*, 2011, pp. 43–52.

[18] C. Ennis, C. Peters, and C. O'Sullivan, "Perceptual Effects of Scene Context and Viewpoint for Virtual Pedestrian Crowds," *ACM Trans. Appl. Percept.*, vol. 8, no. 2, pp. 10:1–10:22, 2011.

[19] S. Kim, S. J. Guy, D. Manocha, and M. C. Lin, "Interactive Simulation of Dynamic Crowd Behaviors Using General Adaptation Syndrome Theory," in *I3D*, 2012, pp. 55–62.

[20] A. Golas, R. Narain, and M. Lin, "Hybrid Long-range Collision Avoidance for Crowd Simulation," in *I3D*, 2013, pp. 29–36.

[21] S. Singh, M. Kapadia, P. Faloutsos, and G. Reinman, "SteerBench: a Benchmark Suite for Evaluating Steering Behaviors," *Comp. Anim. Virt. Worlds*, vol. 20, no. 5-6, pp. 533–548, 2009.

[22] E. Ju, M. G. Choi, M. Park, J. Lee, K. H. Lee, and S. Takahashi, "Morphable Crowds," *ACM Trans. Graph.*, vol. 29, no. 6, pp. 140:1–140:10, 2010.

[23] M. Kapadia, M. Wang, S. Singh, G. Reinman, and P. Faloutsos, "Scenario Space: Characterizing Coverage, Quality, and Failure of Steering Algorithms," in *SCA*, 2011, pp. 53–62.

[24] S. R. Musse, V. J. Cassol, and C. R. Jung, "Towards a Quantitative Approach for Comparing Crowds," *Comp. Anim. Virt. Worlds*, vol. 23, no. 1, pp. 49–57, 2012.

[25] D. Wolinski, S. J. Guy, A.-H. Olivier, M. C. Lin, D. Manocha, and J. Pettré, "Parameter Estimation and Comparative Evaluation of Crowd Simulations," *Comp. Graph. Forum*, vol. 33, no. 2, pp. 303–312, 2014.

[26] S. J. Guy, J. van den Berg, W. Liu, R. Lau, M. C. Lin, and D. Manocha, "A Statistical Similarity Measure for Aggregate Crowd Dynamics," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 190:1–190:11, 2012.

[27] A. Lerner, Y. Chrysanthou, A. Shamir, and D. Cohen-Or, "Data Driven Evaluation of Crowds," in *Motion in Games*, 2009, pp. 75–83.

[28] P. Charalambous, I. Karamouzas, S. J. Guy, and Y. Chrysanthou, "A Data-Driven Framework for Visual Crowd Analysis," *Comp. Graph. Forum*, vol. 33, no. 7, pp. 41–50, 2014.

[29] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[30] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical Dirichlet Processes," *J. Am. Stat. Assoc.*, vol. 101, no. 476, pp. 1566–1581, 2006.

[31] H. Wang and C. O'Sullivan, "Globally Continuous and non-Markovian Crowd Activity Analysis from Videos," in *ECCV*, 2016, pp. 527–544.

[32] L. Fei-Fei and P. Perona, "A Bayesian Hierarchical Model for Learning Natural Scene Categories," in *CVPR*, 2005, pp. 524–531.

[33] E. B. Sudderth, A. Torralba, W. T. Freeman, and A. S. Willsky, "Describing Visual Scenes Using Transformed Objects and Parts," *Int J Comput Vis.*, vol. 77, no. 1-3, pp. 291–330, 2007.

[34] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman, "Discovering Object Categories in Image Collections," *ICCV*, 2005.

[35] J. C. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words," *Int. J. Comp. Vision*, vol. 79, no. 3, pp. 299–318, 2008.

[36] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley-Interscience, 2005.

[37] B. Zhou, X. Wang, and X. Tang, "Random Field Topic Model for Semantic Region Analysis in Crowded Scenes from Tracklets," in *CVPR*, 2011, pp. 3441–3448.

[38] X. Wang, X. Ma, and W. Grimson, "Unsupervised Activity Perception in Crowded and Complicated Scenes Using Hierarchical Bayesian Models," *IEEE Trans. Patt. Anal. Machine Intel.*, vol. 31, no. 3, pp. 539–555, 2009.

[39] B. Zhou, X. Wang, and X. Tang, "Understanding Collective Crowd Behaviors: Learning a Mixture Model of Dynamic Pedestrian-agents," in *CVPR*, 2012, pp. 2871–2878.

[40] T. Ikeda, Y. Chigodo, D. Rea, F. Zanlungo, M. Shiomi, and T. Kanda, "Modeling and Prediction of Pedestrian Behavior Based on the Sub-goal Concept," *Robotics*, p. 137, 2013.

[41] S. Ali and M. Shah, "A Lagrangian Particle Dynamics Approach for Crowd Flow Segmentation and Stability Analysis," in *CVPR*, Jun. 2007, pp. 1–6.

[42] J. Zhong, W. Cai, L. Luo, and H. Yin, "Learning Behavior Patterns from Video: A Data-driven Framework for Agent-based Crowd Modeling," in *Autonomous Agents and Multiagent Systems*, 2015, pp. 801–809.

[43] J. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," in *Berkeley Symp. on Math. Statist. and Prob.*, 1967, pp. 281–297.

[44] C. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2007.

[45] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, 2000.

[46] Y. W. Teh, K. Kurihara, and M. Welling, "Collapsed Variational Inference for HDP," in *NIPS*, 2008.

[47] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic Variational Inference," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 1303–1347, 2013.

[48] C. Wang, J. Paisley, and D. M. Blei, "Online Variational Inference for the Hierarchical Dirichlet Process," in *AISTATS*, 2011.

[49] M. Moussad, D. Helbing, S. Garnier, A. Johansson, M. Combe, and G. Theraulaz, "Experimental Study of the Behavioural Mechanisms Underlying Self-organization in Human Crowds," *Proc. Biol. Sci.*, vol. 276, no. 1668, pp. 2755–2762, 2009.

[50] S. Paris, J. Pettré, and S. Donikian, "Pedestrian Reactive Navigation for Crowd Simulation: a Predictive Approach," *Comp. Graph. Forum*, vol. 26, no. 3, pp. 665–674, 2007.

[51] J. v. d. Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-Body Collision Avoidance," in *Robotics Research*, 2011, no. 70, pp. 3–19.

[52] G. Snook, "Simplified 3d Movement and Pathfinding Using Navigation Meshes," in *Game Programming Gems*, M. DeLoura, Ed., 2000, pp. 288–304.

[53] J.-C. Latombe, *Robot Motion Planning*. Norwell, MA, USA: Kluwer Academic Publishers, 1991.

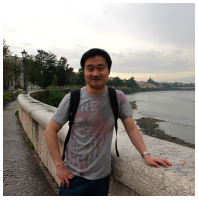
[54] O. Khatib, "Real-time Obstacle Avoidance for Manipulators and Mobile Robots," in *IEEE ICRA*, vol. 2, Mar. 1985, pp. 500–505.

[55] S. Curtis, A. Best, and D. Manocha, "Menge: A Modular Framework for Simulating Crowd Movement," *University of North Carolina at Chapel Hill, Tech. Rep*, 2014.

[56] F. Lamarche and S. Donikian, "Crowd of Virtual Humans: A New Approach for Real Time Navigation in Complex and Structured Environments," *Comp. Graph. Forum*, vol. 23, no. 3, pp. 509–518, 2004.

[57] M. Kapadia, M. Wang, G. Reinman, and P. Faloutsos, "Improved Benchmarking for Steering Algorithms," in *Motion in Games*, 2011, pp. 266–277.

[58] G. Berseth, M. Kapadia, B. Haworth, and P. Faloutsos, "SteerFit: Automated Parameter Fitting for Steering Algorithms," in *SCA*, 2014, pp. 113–122.



He Wang is an Assistant Professor (Lecturer in UK) at School of Computing, University of Leeds, UK. His current research interest is mainly in computer graphics, vision and machine learning and applications. Previously he was a Postdoctoral Associate at Disney Research Los Angeles. He received my PhD in 2012 and did a post-doc afterwards both in School of Informatics, University of Edinburgh, where he mainly worked on motion control of character, deformable object control and 3D scene analysis. Before his PhD, he worked in industry for 4 years after graduating from College of Computer Science and Technology, Zhejiang University, China.

Jan Ondřej is currently Research Fellow at Trinity College Dublin. Previously, he worked as a Postdoctoral Associate at Disney Research Los Angeles and Trinity College Dublin. He obtained his Ph.D. from INSA/INRIA Rennes in France in 2011, supervised by Julien Pettré. His main research interests include crowd simulation, animation, and virtual and augmented reality.

Carol O'Sullivan is the Professor of Visual Computing at Trinity College Dublin, where she has been on the faculty since 1997. From 2013-2016 she was a Senior Research Scientist with Disney Research Los Angeles, and also spent time in the Movement Research Lab in Seoul National University as a Visiting Professor from 2012-2013. Her research interests include perception, animation, virtual humans, and crowds. She was the co-Editor in Chief of the ACM Transactions on Applied Perception, and was formerly Associate EIC of IEEE CG&A. She has served on many editorial boards and program committees, including the SIGGRAPH and Eurographics papers committees, and has been program or general chair for several conferences, e.g., Eurographics, ACM Symposium on Computer Animation, ACM Symposium on Applied Perception, and others.