Path Patterns: Analyzing and Comparing Real and Simulated Crowds

He Wang* Jan Ondřej† Carol O'Sullivan‡
Disney Research Los Angeles

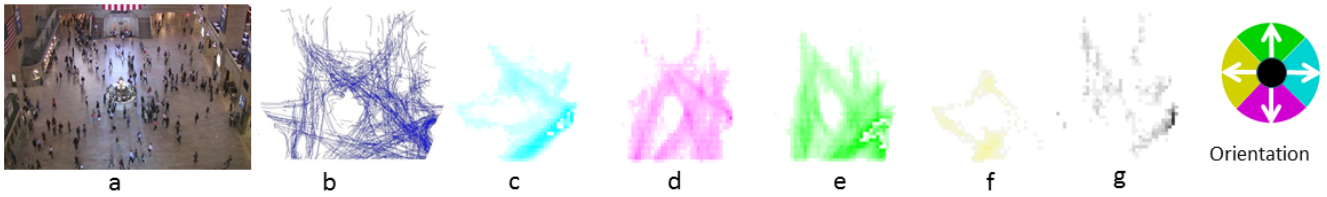

Figure 1: *(a) A video screenshot from a train station; (b) 1000 tracklets (randomly selected from 19999); (c-g) The five orientation subdomains of the top pattern as location-orientation distributions. Inset shows discretization of the orientation, with black representing zero velocity.*

Abstract

Crowd simulation has been an active and important area of research in the field of interactive 3D graphics 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 a new approach based on finding 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's behaviour. To this end, we present a new Stochastic Variational Dual Hierarchical Dirichlet Process (*SV-DHDP*) model. The fidelity of the patterns is then 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.

Keywords: Crowd Simulation, Crowd Comparison, Data-Driven, Clustering, Hierarchical Dirichlet Process, Stochastic Optimization

Concepts: •**Computing methodologies** → **Motion processing;** *Topic modeling;* Bayesian network models;

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.

In this paper, 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. 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. Our approach 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.

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 view-

*realcrane@gmail.com (corresponding author)
†jan.ondrej@disneyresearch.com
‡carol.osullivan@tcd.ie

I3D '16,, February 27-28, 2016, Redmond, WA, USA
ISBN: 978-1-4503-4043-4/16/03
DOI: http://dx.doi.org/10.1145/2856400.2856410...$15.00

ing the original data. Furthermore, we propose a quantitative metric that computes the similarity between both real and 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 [Funge et al. 1999]. To model these myriad behavioral aspects, methods such as field and flow based [Narain et al. 2009; Treuille et al. 2006], force-based [Helbing and Molnár 1995; Karamouzas et al. 2009], velocity and geometric optimization [van den Berg et al. 2008; Pettré et al. 2009; Ondřej et al. 2010] and data-driven [Lee et al. 2007; Lerner et al. 2009b] 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 [Kim et al. 2013; Lemercier et al. 2012] and perceptual experiments [McDonnell et al. 2008; Guy et al. 2011; Ennis et al. 2011]. Quantitative methods fall into two main categories: model-based [Kim et al. 2012; Golas et al. 2013] and data-driven [Singh et al. 2009; Ju et al. 2010; Kapadia et al. 2011; Musse et al. 2012]. Data-driven metrics have been proposed that use the statistics of geometric and dynamic feature analysis [Wolinski et al. 2014], model-based comparisons of motion randomness [Guy et al. 2012] and decision making processes [Lerner et al. 2009a].

Our data-driven evaluation method is partly inspired by two previous approaches. Guy et al. [2012] 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. [2014] 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 [Blei et al. 2003; Teh et al. 2006] have been successfully employed for scene classification [Fei-Fei and Perona 2005; Sudderth et al. 2007], object recognition [Sivic et al. 2005], human action detection [Niebles et al. 2008] and video analysis [Kaufman and Rousseeuw 2005; Zhou et al. 2011; Wang et al. 2009]. 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. [2009] 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 [2012] 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 [2013] 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 [Ali and Shah 2007; Zhong et al. 2015] 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 [MacQueen 1967] and Gaussian Mixture Models (GMMs) [Bishop 2007], require a pre-defined cluster number. Hierarchical Agglomerative Clustering [Kaufman and Rousseeuw 2005] does not require a predefined cluster number, but the user must decide when to stop merging, which is similarly problematic. Spectral-based clustering methods [Shi and Malik 2000] 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 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 $O(n^2)$.

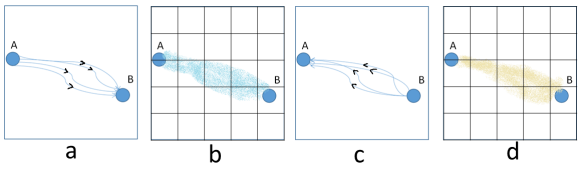

Figure 2: *Two sets of trajectories (a, c) and their corresponding trending paths modeled by Multinomials (b, d). Color coding represents different orientation sub-domains (cf. Figure 1)*

We first define the terminologies in Table 1. Our SV-DHDP model employs a Dual Hierarchical Dirichlet Process, similar to that presented in [Wang et al. 2009], 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

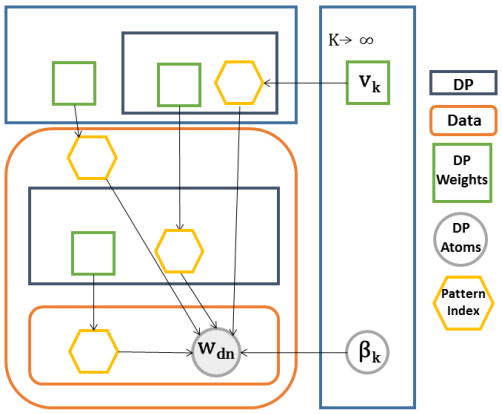

Figure 3: *SV-DHDP model. DP is Dirichlet Process. w_{dn} is the nth agent state in data segment d. K is the total number of patterns. v_k is the weight of the kth pattern. β_k is the kth pattern. Arrows indicate dependencies.*

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*

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.

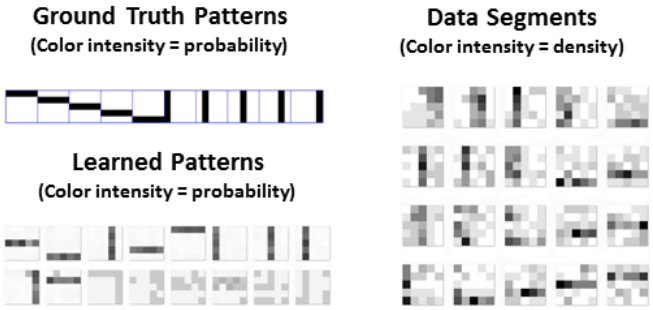

Figure 4: *Illustrative example with 100 data segments each with accumulative 50 positions: (top left) 10 ground truth path patterns; (right) example data; (bottom left) The top 16 path patterns learned*

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 Metric

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 [Teh et al. 2008]. 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 [Teh et al. 2008; Hoffman et al. 2013; Wang et al. 2011]. Our SVI technique is similar to that recently proposed in [Hoffman et al. 2013], 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 math 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 [Hoffman et al. 2013; Teh et al. 2008; Wang et al. 2011] 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 and $\beta_{k,w_j^{ho}}$ is the probability of the state w_j^{ho} in path pattern β_k. 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 a quantitative similarity can be computed for simulation v.s simulation, real data vs. simulation or even real data vs. real data. 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. Another seemingly good idea would be to only compare the top n patterns from both pattern sets. However, it is unfair because the patterns are weighted differently within their sets. And the choice of n is unclear. A more elegant metric is 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 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 [Moussaïd et al. 2009] is a recent version of Helbing's social force model; PETT09 [Pettré et al. 2009] is a velocity obstacle method based approach similar to RVO; ONDREJ10 [Ondřej et al. 2010] uses bearing angle to avoid collisions; and PARIS07 [Paris et al. 2007] 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.

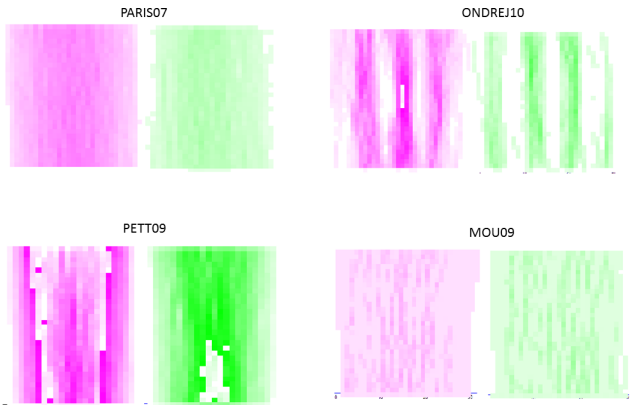

Figure 5: *Top path patterns from the data created by four representative local steering algorithms*

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

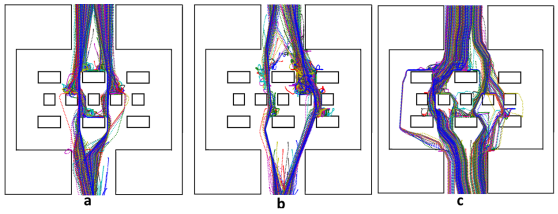

Figure 6: *Trajectories created by three global path planning algorithms: Navmesh, Roadmap and PoField*

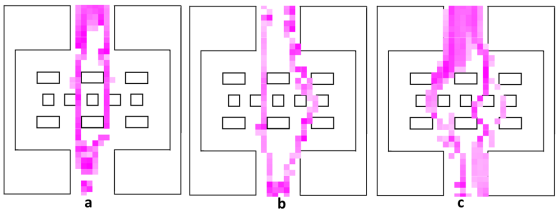

Figure 7: *Top path patterns from the data created by three global path planning algorithms: Navmesh, Roadmap and PoField*

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 [van den Berg et al. 2011] 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 [Snook 2000], Roadmap [Latombe 1991] and Potential Field [Khatib 1985] are used here, referred as Navmesh, Roadmap and PoField. Trajectories are generated using Menge [Curtis et al. 2014] (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 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 Datasets

In addition to simulation data, we also show experimental results computed on two real datasets. 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 [Zhong et al. 2015] (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.

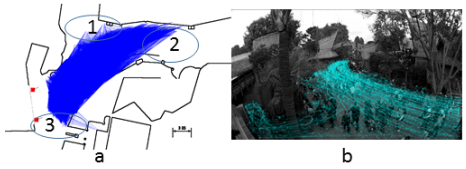

Figure 8: *Park dataset: (a) Projected trajectories, (b) Annotated trajectories overlaid on a frame from the video. Red dots are cameras.*

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 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 Comparison with previous approaches

We empirically compare our SVI method with Gibbs sampling in [Wang et al. 2009] 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 run our method until it converges, then run Gibbs sampling for various times to compare the results.

We first run the sampling for 1.95 hours and show the results in Figure 10. 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 10 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

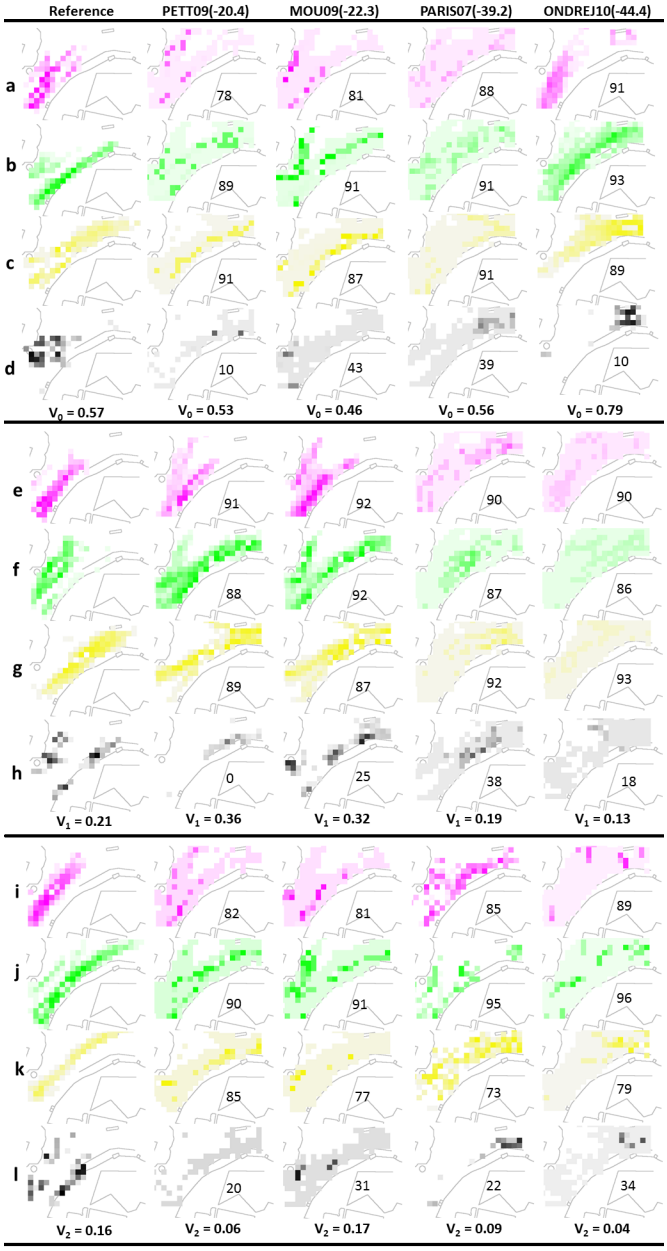

Figure 9: *Top 3 patterns for the Park dataset that cover more than 90% of weights. Each pattern is shown for 4 directions in a group (4 rows – Blue is omitted because no significant pattern found for that direction). Column 1: Top patterns from real dataset. Columns 2-5: Top patterns from four simulated 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.*

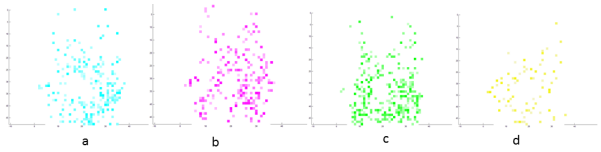

Figure 10: *The top pattern in different velocity domains learned by sampling. More patterns are available in the supplementary material.*

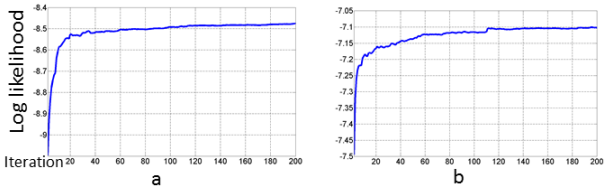

Figure 11: *Model fitness plot with iterations. a: Train Station. b: ONDREJ10 in Section 5.1.1.*

contrast, sampling uses all of them. Figure 11 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

In this section, we show that how our similarity metric can be used to provide meaningful comparisons between real reference data and simulation data.

We used the four models in Section 5.1.1 in combination with one global path planning method [2004] 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 12. Some top patterns are shown in Figure 9 column 2-5 and Figure 12 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

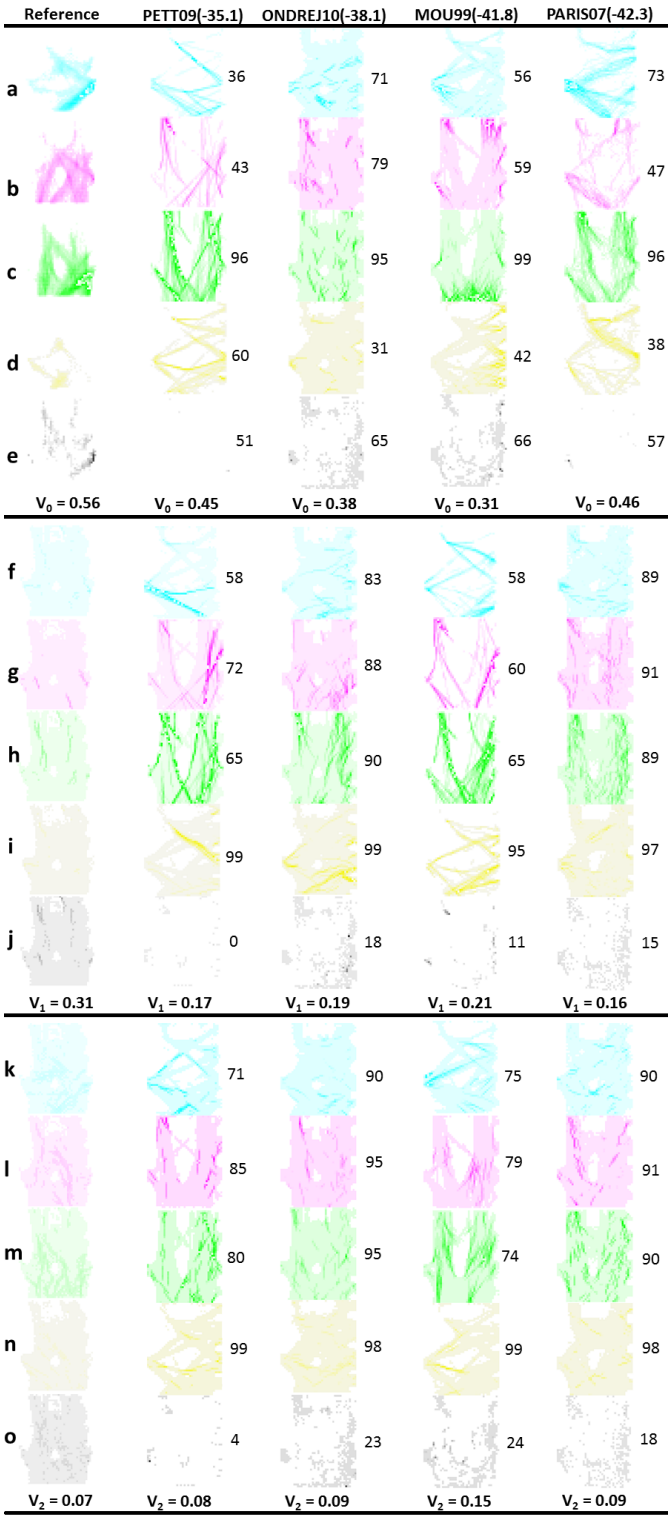

Figure 12: *Train Station patterns for real and simulated datasets. The layout and scores are computed in the same way as in Figure 9.*

path planning has a great influence. One example is Figure 12 (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 12 (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 12, 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 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 12, 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.

7 Conclusions and Future Work

We propose a new perspective for comparing crowd data. We present a non-parametric hierarchical Bayesian model to automatically extract a desirable set of patterns. Also, we propose a similarity metric for comparison.

Our metric is environment-based. The main reason is the reference data is almost always affected by the environment in real-world applications. When only local flow patterns are needed, our metric still works well as shown in the bidirectional flow example. A global shift of flows will give low scores but we argue that they can always be aligned if a rotation/translation invariant comparison is needed.

Our method has some inherent limitations. Firstly, our method does not directly measure individual trajectories thus does not reflect individual visual similarities. Our patterns are reflections of information on a higher level than individual trajectories. Secondly, it does not capture temporal information such as changes of patterns over time. Lastly, our truncation-based stochastic variational inference is sensitive to the initialization even if the stochasticity in gradient helps to some extent. In our experiments, we did grid search to find out good initializations.

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 [Berseth et al. 2014]. Currently, we only capture flows;although individual trajectories may also influence perceptual realism. 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 so that it becomes a behavior pattern model. We believe it will further help simulating realistic crowds with diverse behaviors.

References

ALI, S., AND SHAH, M. 2007. A Lagrangian Particle Dynamics Approach for Crowd Flow Segmentation and Stability Analysis. In *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR '07*, 1–6.

BERSETH, G., KAPADIA, M., HAWORTH, B., AND FALOUTSOS, P. 2014. SteerFit: Automated Parameter Fitting for Steering Algorithms. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '14, 113–122.

BISHOP, C. 2007. *Pattern Recognition and Machine Learning*. Springer, New York.

BLEI, D. M., NG, A. Y., AND JORDAN, M. I. 2003. Latent Dirichlet Allocation. *J. Mach. Learn. Res. 3*, 993–1022.

CHARALAMBOUS, P., KARAMOUZAS, I., GUY, S. J., AND CHRYSANTHOU, Y. 2014. A Data-Driven Framework for Visual Crowd Analysis. *Comp. Graph. Forum 33*, 7, 41–50.

CURTIS, S., BEST, A., AND MANOCHA, D. 2014. Menge: A modular framework for simulating crowd movement. *University of North Carolina at Chapel Hill, Tech. Rep*.

ENNIS, C., PETERS, C., AND O'SULLIVAN, C. 2011. Perceptual Effects of Scene Context and Viewpoint for Virtual Pedestrian Crowds. *ACM Trans. Appl. Percept. 8*, 2, 10:1–10:22.

FEI-FEI, L., AND PERONA, P. 2005. A Bayesian hierarchical model for learning natural scene categories. In *IEEE CVPR 2005*, 524–531.

FUNGE, J., TU, X., AND TERZOPOULOS, D. 1999. Cognitive Modeling: Knowledge, Reasoning and Planning for Intelligent Characters. In *SIGGRAPH'99*, ACM Press/Addison-Wesley Publishing Co., 29–38.

GOLAS, A., NARAIN, R., AND LIN, M. 2013. Hybrid Long-range Collision Avoidance for Crowd Simulation. In *I3D 2013*, 29–36.

GUY, S. J., KIM, S., LIN, M. C., AND MANOCHA, D. 2011. Simulating Heterogeneous Crowd Behaviors Using Personality Trait Theory. In *SCA 2011*, 43–52.

GUY, S. J., VAN DEN BERG, J., LIU, W., LAU, R., LIN, M. C., AND MANOCHA, D. 2012. A Statistical Similarity Measure for Aggregate Crowd Dynamics. *ACM Trans. Graph. 31*, 6, 190:1–190:11.

HELBING, D., AND MOLNÁR, P. 1995. Social force model for pedestrian dynamics. *Phys. Rev. E 51*, 5, 4282–4286.

HOFFMAN, M. D., BLEI, D. M., WANG, C., AND PAISLEY, J. 2013. Stochastic Variational Inference. *J. Mach. Learn. Res. 14*, 1, 1303–1347.

IKEDA, T., CHIGODO, Y., REA, D., ZANLUNGO, F., SHIOMI, M., AND KANDA, T. 2013. Modeling and prediction of pedestrian behavior based on the sub-goal concept. *Robotics*, 137.

JU, E., CHOI, M. G., PARK, M., LEE, J., LEE, K. H., AND TAKAHASHI, S. 2010. Morphable Crowds. *ACM Trans. Graph. 29*, 6, 140:1–140:10.

KAPADIA, M., WANG, M., SINGH, S., REINMAN, G., AND FALOUTSOS, P. 2011. Scenario Space: Characterizing Coverage, Quality, and Failure of Steering Algorithms. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, New York, NY, USA, SCA '11, 53–62.

KARAMOUZAS, I., HEIL, P., BEEK, P. V., AND OVERMARS, M. H. 2009. A Predictive Collision Avoidance Model for Pedestrian Simulation. In *Motion in Games*, 41–52.

KAUFMAN, L., AND ROUSSEEUW, P. J. 2005. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley-Interscience.

KHATIB, O. 1985. Real-time obstacle avoidance for manipulators and mobile robots. In *1985 IEEE International Conference on Robotics and Automation. Proceedings*, vol. 2, 500–505.

KIM, S., GUY, S. J., MANOCHA, D., AND LIN, M. C. 2012. Interactive Simulation of Dynamic Crowd Behaviors Using General Adaptation Syndrome Theory. In *I3D 2012*, 55–62.

KIM, S., GUY, S. J., AND MANOCHA, D. 2013. Velocity-based Modeling of Physical Interactions in Multi-agent Simulations. In *SCA 2013*, 125–133.

LAMARCHE, F., AND DONIKIAN, S. 2004. Crowd of virtual humans: a new approach for real time navigation in complex and structured environments. *Computer Graph. Forum 23*, 3, 509–518.

LATOMBE, J.-C. 1991. *Robot Motion Planning*. Kluwer Academic Publishers, Norwell, MA, USA.

LEE, K. H., CHOI, M. G., HONG, Q., AND LEE, J. 2007. Group Behavior from Video: A Data-driven Approach to Crowd Simulation. In *SCA 2007*, Eurographics Association, 109–118.

LEMERCIER, S., JELIC, A., KULPA, R., HUA, J., FEHRENBACH, J., DEGOND, P., APPERT-ROLLAND, C., DONIKIAN, S., AND PETTRÉ, J. 2012. Realistic Following Behaviors for Crowd Simulation. *Comp. Graph. Forum 31*, 2, 489–498.

LERNER, A., CHRYSANTHOU, Y., SHAMIR, A., AND COHEN-OR, D. 2009. Data Driven Evaluation of Crowds. In *Motion in Games*. 75–83.

LERNER, A., FITUSI, E., CHRYSANTHOU, Y., AND COHEN-OR, D. 2009. Fitting Behaviors to Pedestrian Simulations. In *SCA 2009*, 199–208.

MACQUEEN, J. 1967. Some methods for classification and analysis of multivariate observations. In *Berkeley Symp. on Math. Statist. and Prob.*, 281–297.

MCDONNELL, R., LARKIN, M., DOBBYN, S., COLLINS, S., AND O'SULLIVAN, C. 2008. Clone Attack! Perception of Crowd Variety. *ACM Trans. Graph. 27*, 3, 26:1–26:8.

MOUSSAÏD, M., HELBING, D., GARNIER, S., JOHANSSON, A., COMBE, M., AND THERAULAZ, G. 2009. Experimental study of the behavioural mechanisms underlying self-organization in human crowds. *Proc. Biol. Sci. 276*, 1668, 2755–2762.

MUSSE, S. R., CASSOL, V. J., AND JUNG, C. R. 2012. Towards a Quantitative Approach for Comparing Crowds. *Comp. Anim. Virt. Worlds 23*, 1, 49–57.

NARAIN, R., GOLAS, A., CURTIS, S., AND LIN, M. C. 2009. Aggregate Dynamics for Dense Crowd Simulation. *ACM Trans. Graph. 28*, 5, 122:1–122:8.

NIEBLES, J. C., WANG, H., AND FEI-FEI, L. 2008. Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words. *Int. J. Comp. Vision 79*, 3, 299–318.

ONDŘEJ, J., PETTRÉ, J., OLIVIER, A.-H., AND DONIKIAN, S. 2010. A Synthetic-vision Based Steering Approach for Crowd Simulation. *ACM Trans. Graph. 29*, 4, 123:1–123:9.

PARIS, S., PETTRÉ, J., AND DONIKIAN, S. 2007. Pedestrian Reactive Navigation for Crowd Simulation: a Predictive Approach. *Comp. Graph. Forum 26*, 3, 665–674.

PETTRÉ, J., ONDŘEJ, J., OLIVIER, A.-H., CRETUAL, A., AND DONIKIAN, S. 2009. Experiment-based Modeling, Simulation and Validation of Interactions Between Virtual Walkers. In *SCA 2009*, ACM, 189–198.

SHI, J., AND MALIK, J. 2000. Normalized Cuts and Image Segmentation. *IEEE Trans. Patt. Anal. Mach. Intell. 22*, 8, 888–905.

SINGH, S., KAPADIA, M., FALOUTSOS, P., AND REINMAN, G. 2009. SteerBench: a benchmark suite for evaluating steering behaviors. *Comp. Anim. Virtual Worlds 20*, 5-6, 533–548.

SIVIC, J., RUSSELL, B. C., EFROS, A. A., ZISSERMAN, A., AND FREEMAN, W. T. 2005. Discovering object categories in image collections. *ICCV 2005*.

SNOOK, G. 2000. Simplified 3d Movement and Pathfinding Using Navigation Meshes. In *Game Programming Gems*, M. DeLoura, Ed. Charles River Media, 288–304.

SUDDERTH, E. B., TORRALBA, A., FREEMAN, W. T., AND WILLSKY, A. S. 2007. Describing Visual Scenes Using Transformed Objects and Parts. *Int J Comput Vis 77*, 1-3, 291–330.

TEH, Y. W., JORDAN, M. I., BEAL, M. J., AND BLEI, D. M. 2006. Hierarchical Dirichlet Processes. *J. Am. Stat. Assoc. 101*, 476, 1566–1581.

TEH, Y. W., KURIHARA, K., AND WELLING, M. 2008. Collapsed Variational Inference for HDP. In *NIPS 2008*.

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

VAN DEN BERG, J., LIN, M., AND MANOCHA, D. 2008. Reciprocal Velocity Obstacles for real-time multi-agent navigation. In *IEEE International Conference on Robotics and Automation, 2008. ICRA 2008*, 1928–1935.

VAN DEN BERG, J., GUY, S. J., LIN, M., AND MANOCHA, D. 2011. Reciprocal n-Body Collision Avoidance. In *Robotics Research*, C. Pradalier, R. Siegwart, and G. Hirzinger, Eds., no. 70 in Springer Tracts in Advanced Robotics. Springer Berlin Heidelberg, 3–19. DOI: 10.1007/978-3-642-19457-3_1.

WANG, X., MA, X., AND GRIMSON, W. 2009. Unsupervised Activity Perception in Crowded and Complicated Scenes Using Hierarchical Bayesian Models. *IEEE Trans. Patt. Anal. Machine Intel. 31*, 3, 539–555.

WANG, C., PAISLEY, J., AND BLEI, D. M. 2011. Online variational inference for the hierarchical Dirichlet process. In *AISTATS*.

WOLINSKI, D., GUY, S. J., OLIVIER, A.-H., LIN, M. C., MANOCHA, D., AND PETTRÉ, J. 2014. Parameter estimation and comparative evaluation of crowd simulations. *Comp. Graph. Forum 33*, 2, 303–312.

ZHONG, J., CAI, W., LUO, L., AND YIN, H. 2015. Learning Behavior Patterns from Video: A Data-driven Framework for Agent-based Crowd Modeling. In *Autonomous Agents and Multiagent Systems*, 801–809.

ZHOU, B., WANG, X., AND TANG, X. 2011. Random field topic model for semantic region analysis in crowded scenes from tracklets. In *IEEE CVPR 2011*, 3441–3448.

ZHOU, B., WANG, X., AND TANG, X. 2012. Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, IEEE, 2871–2878.