SPS Chapter Research Award Proposal

Quantitative evaluation of pedestrian movement models:
A *real* many-body problem

School:	Purdue University
Chapter:	5781
Requested:	$1089.00
Project Leader:	Dawith Lim

Abstract

A number of papers have been published since 2000 which attempt to model pedestrian crowd flow dynamics using basic equations of motion. Here we propose a study which aims to collect research grade pedestrian data on campus using an off-the-shelf aerial drone, and test a pedestrian dynamics model on the data.

1 Summary of research activity

1.1 Objectives

There were three main objectives that have been identified for this project:

1. Train undergraduates in modern research in physics, incorporating contemporary technological advances readily available.
2. Examine aerial drone footage as a viable means of collecting research-grade data of natural pedestrian behaviors.
3. Performing replication study and evaluating various models in existing literature using our data.

1.2 Raw data collection

We have collected aerial footages of pedestrian movement using the DJI Mavic pro purchased with the budget. Including flight control practices, a total of fifteen flights were performed using the drone, nine of which were conducted with the intent of collecting data, and three of the raw data have been processed for the purpose of demonstration.

Initially, the data was taken from a variety of heights, ranging from as low as 25 m and as high as 70 m above ground level. To avoid alerting the drone's presence (and thereby affecting pedestrian behavior) and to have a good balance between adequate amount of features for detection and scope of detection, the recording height was fixed to 35 m above ground level for the subsequent data collections. Also, the initial data were taken from four different locations on campus, but two of these locations were dropped from subsequent recordings because there were too many trees obstructing the drone's field of view.

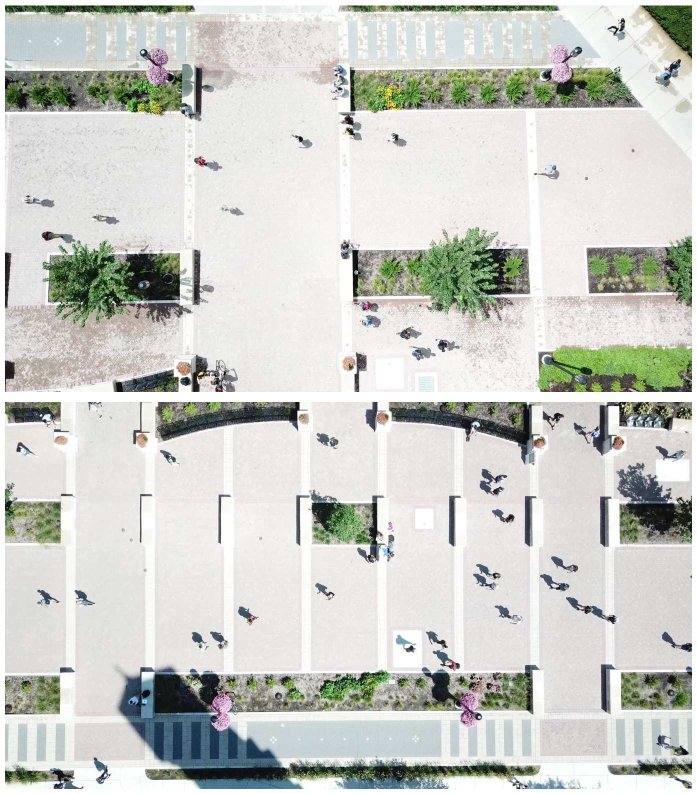

Figure 1: Sample frames from two datasets, A (left) and B (right), taken on Purdue campus.

As seen in figure 1, the drone footage boasts a superb quality. Part of the justification we presented for purchasing DJI Mavic pro was its image stability, and the drone delivered the stability it promised, as we did not have to deal with linear and angular drift at all. If these issues were present (especially in combination), we would have had to add another step in data processing and add to the already high computational workload.

1.3 Data processing

The primary technical challenge associated with the project was translating the raw footage collected by the drone into identified trajectories to be run. This was broken down into two problems: human detection from birds-eye view, and single-particle tracking. Once trajectories are collected, they were loaded into a JSON format. JSON was selected because it can easily be used in any programming environment, has high human readability and easily accommodates trajectories of various lengths.

Birds eye human detection was solved with machine learning; however, several factors complicated straightforward application of a sliding window classifier. First, the large 4K resolution of the DJI Mavic Pro Drone made the runtime on a sliding classifier prohibitive; despite being configured for GPGPU (General-Purpose computation on Graphic Processing Units) on a fairly high-powered GPU (Nvidia GTX 1080), the process took up to 23 hours to execute on a ∼5 minutes long footage. It also made single-shot localizers difficult, as the ratio between the size of the whole image and the people was so large that their smaller features would become negligible in final stage detections. Secondly, the angle of the camera view, (from directly above) was sufficiently unique that our group could not find an adequate existing dataset for training. Usage of the general purpose PASCAL

VOC 2007 - 2012 dataset was attempted, as it does contain significant numbers of tagged Person labels from different angles, however the large size of the input frames created significant numbers of false positives.

To resolve the first issue, the field of view was split into sections, and then the outputs for all of the sections were stitched together after processing. The single shot detector selected was YOLO v2 in a single class configuration. YOLO v2 was selected because of its runtime was faster than other alternatives while maintaining a reasonable accuracy. Moreover, its widespread usage gave the group access to existing open-source implementations. The stitching process was done by collecting all bounding boxes that existed on the edge of the single-shot interfaces and calculating a likelihood of between all boxes of being created by the same person. This was done with a loss function which penalized distance, as well as how rectangular the box created would be if the two candidate boxes were combined. Boxes were combined by taking the lowest points and highest points to form a new box. This has led to the issue where image detection fails on the boundary and caused segmented trajectories. Future implementations could have overlapping detectors instead of segmented ones for more reliable interface detection, but this improvement could not be implemented due to time constraints.

The absence of available training dataset was solved through the manual tagging of collected data. Footage of students walking between passing periods was collected, and was used as raw data for SPS club members to manually tag. A web interface was created to more easily decentralize the process of data tagging, hosting the web server on Purdues local network. This allowed many people to tag humans in an image simultaneously, from anywhere on campus. A dataset of people viewed from above was created, of 5048 images, with none to multiple people in each image. Taggers were told to click directly on the head of the people detected, and that partial people on the edge of the frame were still valid. This method could only account for false negatives; to reduce false positives, the model had to be re-trained on just the initial dataset, and then run on frames with no people. The detections were added to a classified set of non-people. Also, all detections on the raw datasets were run, and different web interface was created for users to distinguish between true and false detections. Those second pass tags were then passed into corresponding datasets, and the model was retrained. This process was repeated twice to reduce the false positives down to an acceptable level.

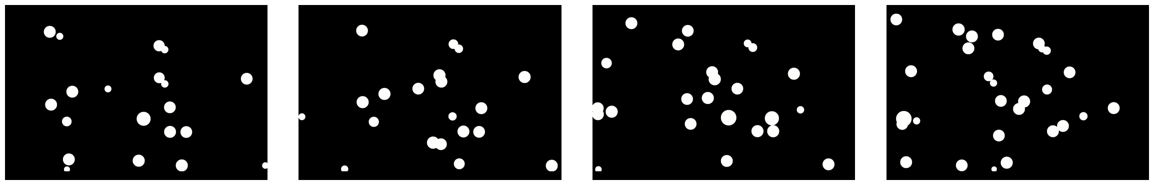

Figure 2: Extracted position data plotted as white circles on black background, before particle tracing step. The four plots (left to right) are taken across 7 seconds of footage.

Figure 2 depicts what our position data looks like. The variations in circle sizes relate to confidence of person detection, which is deemed unimportant once we filter out erroneous detections and form the trajectories. The final product of the data processing yields a list of timestamped trajectories:

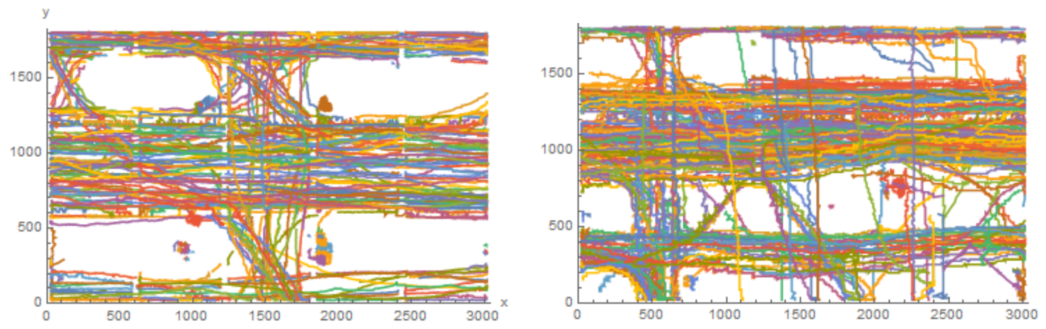

Figure 3: Two complete sets of trajectories produced using the steps above. The trajectory to the left (corresponds to data A) was plotted before segment stitching, and the trajectory to the right (corresponds to data B) was plotted after.

From figure 3, we can conclude that the segment stitching using extrapolation is quite successful. This, of course, is not without problems altogether- for example, when two trajectories happen to segment simultaneously in close vicinity, the two trajectories that are supposed to be distinct are stitched together, forming unrealistic trajectories. Thus, among 700 uniquely labeled trajectories present on each data set, we have manually identified complete trajectories, and ran predictions of those pedestrians only. This still results in more than 100 actively tracked pedestrians per data set. However, all pedestrian trajectories are considered for calculating the inter-pedestrian forces.

The combination of these two solutions allowed us to detect humans with our drone in a reasonable runtime. However, this solution does not work in high density applications, due to the limited number of bounding boxes YOLO can output for a give image. A possible future avenue of experimentation for detection of individuals in high density crowd flow is the usage of a large proposal network for limiting the space in which a more detailed sliding window detector can run. For this paper, the raw data collected was only run on spaces in which individuals have reasonable distance between each other, and the model struggled with people who walked to closely together.

Once detections were collected, single-particle tracking was run. This is a reasonably solved problem, with heavy applications in protein tracking in biophysics, so many libraries exist to generate trajectories, agnostic to how the detections themselves were created. Our group used trackpy, an open source and easy to use tracking toolkit. For particle prediction we assumed constant velocity between two frame, which was reasonable at 30 fps, and removed trajectories that only existed for two seconds. Trackpy handled the trajectory generation, and we reformated them into timestamps and trajectories at a given timesteps.

1.4 Evaluation of pedestrian models

Once we began our work on the papers, we quickly found that most of the papers offered very little description of how their models were implemented computationally. All the papers presented the basic equations, but the algorithms presented only offered a very general outline. The details of many aspects of the models were simply omitted - for example, how the models set the pedestrians' directional inclination (e.g. destination), and how the models defined the obstacles. In the case of pedestrians' desired direction, there were a few papers [3] that did provide some justified explanation on how the destination is defined, and we assumed other models would apply a similar method.

In the case of the obstacles, we could only speculate how the papers handled obstacle detection; all the papers that involves obstacles used pedestrian movements in confined spaces like a corridor or a classroom, where the boundaries of the space naturally forms an impenetrable barrier. In this case, the position of obstacles are given by straight lines from pedestrian to any of the boundaries.

However, in our 'real world' data, there are many obstacles scattered throughout the image, making obstacle specification difficult. Because we were unable to come up with a robust automated method of obstacle detection, we had to create a table of coordinate values along the edges of each obstacles and used this table to find the obstacle nearest to a given pedestrian. Moreover, we noticed that the simple obstacle/no obstacle distinction implicit in all of the models was rather misleading. There are many field objects that people typically avoided, such as a stone bench or a patch of grass, that are nonetheless passable and, in fact, trespassed by a few pedestrians in the observed data.

Due to many issues such as the difficulty in reproducing accurate models, ambiguities and missing information, as well as severely restricted available manpower and computing resources to write, debug and proofread the codes, we were unable to conduct a satisfactorily thorough and conclusive evaluations of the models we have studied. Instead, we have altered our initial plan as to gesture, with a proof-of-concept example, how we would have gone about testing the models. For this example, we have used a lattice gas model adaptation of Helbing's simple model [5][3], coded on Mathematica 11.2.

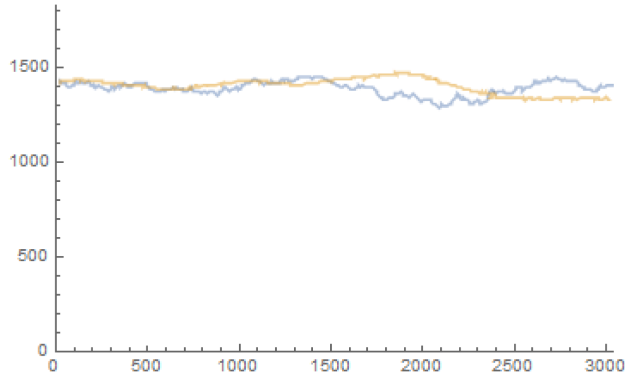

Figure 4: Comparison of one pedestrian trajectory (orange) compared to one iteration of the predicted path (blue).

The prediction successfully follows the overall direction of the actual trajectory, but differs from it in specific response to surroundings as the pedestrian progresses.

To perform the error analysis, the model was run for n consecutive steps from an initial position of a trajectory at some given time, and found the difference from the "actual" position n steps after the initial position. Since the lattice gas model is a probabilistic model, this was repeated 30 times for each initial position and value of n.

The following figure presents the average n^{th} step error from each initial positions along the trajectory.

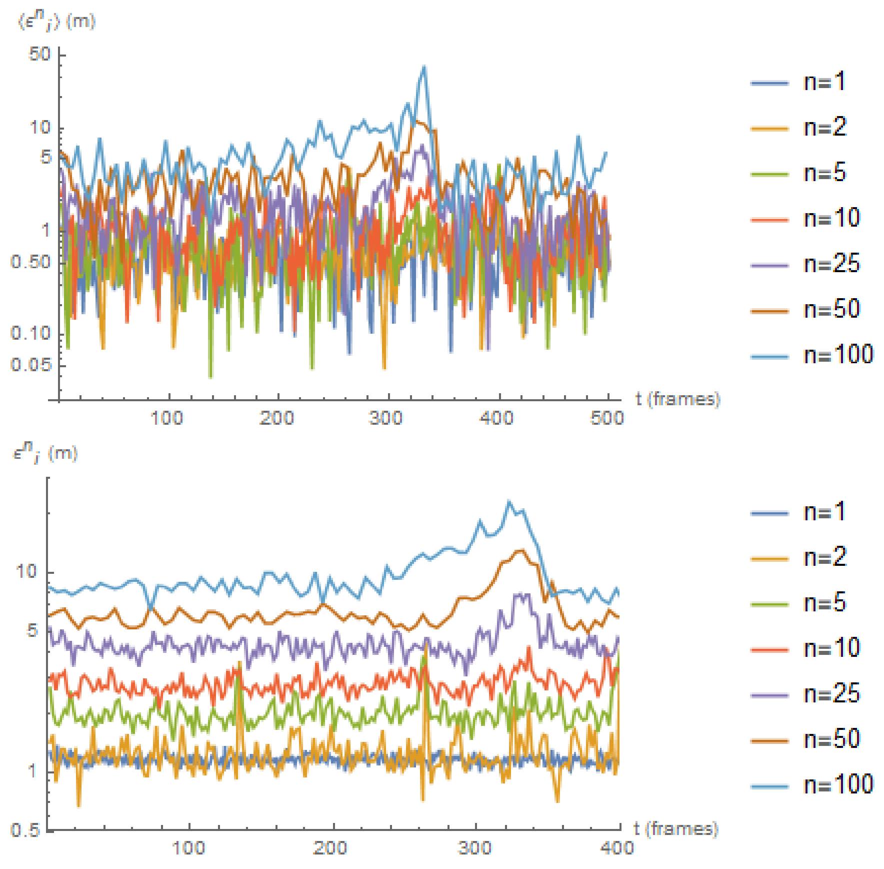

Figure 5: The $\langle \varepsilon_i^n \rangle$ step error and $\left\langle (\varepsilon_i^n)^2 \right\rangle$ step error for one pedestrian. One frame corresponds to 1/30 seconds. (30 fps)

The absolute value of error seems rather large. On the lower value of n, the error is much smaller. It is also worth noting that the scale of the image space is also very large (3040x1824 pixels, or about 280x170m), and so the error relative to the scale is not as big. Finetuning the parameters further could also help improve the accuracy, but this was not completed due to the large runtime required for a thorough testing.

1.5 Group participation

One of the key aims of the project was to test various pedestrian models coded in python, using the real life data collected. Initially, the group's goal was to teach underclassmen how to write models in python, and several attempts were made towards putting together python workshops and coding practices. However, the member participation has seen a large attrition, mostly due to loss of interest, and extremely sharp learning curve. The latter seem to have had a large bearing on participation rate, as members showed active interest in relatively simpler tasks such as manually generating training data or operating the drone. Therefore, this could be attributed to insufficient active guidance and preparation on the upperclassman's part. By September, only two members, Dawith Lim and Charles Li, were regularly participating. This loss of manhours put a great restriction on the amount of work that have been achieved.

That being said, In the early phase of the project where the member participation was still reasonable, the underclassmen learned how to code various simple models of physics, such as the trajectory motion, using python. Most of the footages were also filmed with the help of some underclassmen, who volunteered their time to practice and perform controlled flight according to our data collection protocol. Moreover, each week at the project meeting, the group read one paper related to the topic, where upperclassmen offered a summary and explanations of the paper, and the underclassmen would raise questions as they found necessary. The general feedback from the underclassmen were largely positive. Lastly, during the initiative to produce training data for the drone, about eight people in total participated, which allowed us to obtain more than five thousand individual data points which were subsequently used for training.

1.6 Evaluation

First, we think we have achieved partial success on the educational goal. Although we did not manage to achieve a prolonged participation of most members, we were able to expose the fellow undergraduates, especially the underclassmen, on some simple tasks in hands-on research, and learned how to integrate and develop their knowledge and skills to achieve a research goal. They were also exposed to much of the literature in an exotic topic in soft matter research, which, traditionally, is not something that students are familiar with in their early years in college. It has also taught us, the organizing members, a strong lesson that we cannot expect others to follow by example if we don't offer well planned and coordinated guidance. Should the Purdue chapter of SPS participate in another project under SPS national, we should make sure to devote considerable amount of time on teambuilding to ensure the participating members are well coordinated and motivated throughout the project.

Our second objective, namely, to evaluate whether aerial drone footage could be a viable tool in pedestrian research, was a great success. We have successfully demonstrated that a research-grade empirical pedestrian data can be collected by a combination of off-the-shelf commercial drone with no additional modification, and modern computational tools enabled by recent development of machine learning. Although there are several legal restrictions that apply to drones (e.g. requirement of permission when flying in crowded cities), we believe that the data collection method we have developed is nonetheless viable for most purposes.

Lastly, our attempt at evaluating existing pedestrian models fell short of our goal. Although we were able to demonstrate our intended method of error evaluation, due to limited computational and manpower available at hands, our tests by no means offered a conclusive evidence for or against the models we sought to test. We have, however, managed to gesture how future development in mathematical models of pedestrian dynamics may be tested using empirical data such as those that we have collected.

1.7 Budget

Item	Cost
DJI Mavic Pro quadcopter drone	\$ 1089.00
Replacement propellers	\$ 36.00
Replacement gimbal mount	\$ 6.00
Total	\$ 1131.00

The grant from SPS national was spent on purchasing the drone. The need for replacement parts arose due to flight accidents, and the cost for the parts was covered by the club's own funds.

References

[1] Dan Allan Casper Van Der Wel and Thomas A Caswell. soft matter trackpy github repository. `https://github.com/soft-matter`, 2018.

[2] experiencor. keras-yolo2 github repository. `https://github.com/experiencor/keras-yolo2`, 2018.

[3] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995.

[4] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. *arXiv preprint*, 2017.

[5] Weiguo Song, Xuan Xu, Bing-Hong Wang, and Shunjiang Ni. Simulation of evacuation processes using a multi-grid model for pedestrian dynamics. *Physica A: Statistical Mechanics and its Applications*, 363(2):492–500, 2006.