Stadium Evacuation Simulation

CSE 6730 – Spring 2017 Project 3

Jing Gu, Xun Cao, Xian Li, Zhijie Xi, Shengji Ye
Georgia Institute of Technology
Atlanta, GA 30332, USA
jgu47, xcao65, xli684, zxie62, sye43@gatech.edu

***Abstract*—In this project, a model to simulate the evacuation from Bobby Dodd stadium is built. The pedestrian movement is simulated by automata model, which is stochastic process. Evacuation efficiency is analyzed. The model can be applied to other event evacuation simulation.**

Keywords—Evacuation, Modeling, Cellular Automata

I. INTRODUCTION

In our simulation project 3, we construct a model to simulate the evacuation of Bobby Dodd stadium after an event such as a football game, concert, etc. The system under investigation is the surrounding area of Bobby Dodd stadium in the span of time immediately following an event. The purpose of the simulation is to model the behavior and disbursement of pedestrians from the stadium and in the surrounding area, and then minimize the evacuation time after an event. Benefits of minimizing the evacuation time include increased satisfaction and safety for the attendees of the event. It also reduces the event host cost for Georgia Tech.

We consider that pedestrians may enter the system under investigation from any of the ten stadium gates, and have a desired destination that will not change from this point. We derive estimates of where people are trying to walk to after the game. [1]

We use a cellular automata model for pedestrian movement. So the data is generated based on this model. The simulation is stochastic process, which means we use random numbers to model unpredictable or unknown elements of the system.[2]

We use Google Maps API to get the geographical information of the stadium and its surrounding areas to refine our simulation. We limit the geographical scale of the system to include Georgia Tech's campus and surrounding areas due to the fact that the university is not concerned with off-campus traffic.

II. LITERATURE REVIEW

In *C.Burstedde et al.* 's paper[3], a "floor field" is used to describe short term and long term behavior for pedestrians with varying velocities on a two-dimensional grid. Individual pedestrian is also given a velocity parameter that determines the number of cells moved in a given time step, and a direction parameter that determines their preferred directions. They also offer several quantitative rules such as a 1.3 m/s average speed for pedestrians, and a 40 x 40 cm^2/person space requirement that are generally useful for pedestrian modeling. And based on the speed of pedestrians and the area each pedestrian occupies, it is easy to get the relationship between the simulation timestep and the real time scale.

The main idea of a "floor field" is an underlying grid that contains static features (such as terrain) and dynamic features (such as group interaction). The static features are unique to the specific simulation system, but the dynamic features are the same for all pedestrian behavior. The group interaction in this case is modeled by exponential decay, resulting in movement toward a group being more highly probable. This modeling idea is validated by the fact that commonly encountered behavior patterns that occur in pedestrians such as self-organization into groups and lines and line counter flow in crowded areas are reported to occur.

The above mathematical model is continually modified in the work of *C. Burstedde et al*. In general, the individual pedestrian movement is obtained by the following rules on transition to a target cell based off of the highest neighborhood probability pij. This transition in turn then updates the dynamic floor field.

1) If the target is occupied, a pedestrian will not move here.
2) If the target is empty and no one else plans to move to there, then the particle will move there.
3) If more than one particles plans to move to one target, the particle with the highest probability will move there.

Compared to the simulation that is described, Changes that would be made in our simulation are the preferred direction and “floor field” implementation.

Our preferred direction should instead be calculated at each time step based on the trajectory to a certain destination. Our floor field implementation could also be more efficient by adding additional parameters to each cell in a single grid rather than creating multiple grids. Considering the difference between our project requirement and C. Burstedde et al.’s work, It is necessary to do some modification on the dynamic floor field. Dirk Helbing et al. have presented work on how to deal with different kinds of pedestrians in the simulation model based on what they call a social force model. We plan to incorporate this feature into our dynamic field. The main idea of this model is considering two different situations: normal and panic. It is intuitive because it is possible that some urgent accidents will happen, such as in the crowded church and stadium exit. Then this model treat two different situations with different rules. Shigeyuki Okazakia and Satoshi Matsushitaa summarized the different kinds of queues which can be used in pedestrian simulation and their work may be useful for our pedestrian simulation project. From their work, different situations can be classified into three types of queue behavior based on the observation of the pedestrian’s movement in airport, railway station, department store and office buildings, etc.

III. Model

Our conceptual model is based on information found in literature review. Key entities in the model are pedestrians, entries, exits, which are all endogenous. The surrounding terrain is considered as an entity which is exogenous, because it interacts with pedestrians but not modelled explicitly. The surrounding area of Bobby Dodd stadium will be considered as the area approximately encapsulated by Ferst Drive, North Avenue NW, and West Peachtree Street NW. It is represented by a two dimensional cellular automata grid. Each cell in the grid contains only one person, having a size of 40 x 40 cm. Each populated cell represents a person who has a static destination that they will move toward.[4]

Pedestrian movement is defined as a transition into a neighboring cell in a 3x3 neighborhood grid. This configuration is a relatively standard implementation of cellular automata. Pedestrian in the center cell has 8 choices to move.

The simulation will begin with pedestrians entering the system after the end of an event inside stadium. Because every person’s arrival time is generally considered to be stochastic, the entry of pedestrians into the system is considered to be a semi-stochastic process that is both nonstationary and autonomous. The number of the pedestrians entering into the system has a relationship with time, however, the accurate number is generated stochastically. It is noted that event attendees usually travel in groups. So we use a dynamic floor field to enforce group dynamics, which will make independent pedestrians appear into group. Pedestrians will enter the system form one of the ten gates of the stadium. Each of them is represented as a terrain feature in the static floor field. The number of pedestrians inside the stadium will decrease as every new pedestrian generates at the entry. In the end, the process will terminates when all pedestrians inside the stadium have exited.

The movement of the pedestrians in the system is also stochastic, but highly dependent and stationary.[5] For each populated cell, the pedestrian will move to the highest probability non occupied cell, and in the event of ties between pedestrians, the pedestrian with the highest probability will move to the cell in question. The probability of moving will be will be assigned by the equation used by C. Burstedde et al. shown in the literature review. The direction of movement with the highest probability corresponds neighbor with the least distance to the pedestrian’s desired destination. This distance is calculated by performing a breadth first grid search for every point in the grid that is not blocked.

The arriving of destination event triggers the exit of simulation for the pedestrians, which means a destination entity is needed. Each destination should be assigned with lopsided probability according to its location, direction and property. For instance, it’s reasonable that Midtown Marta station has twice the number of attendees than a single on campus apartment building and people are more likely to get to parking lot rather than the library. So, according to a survey of distributions of people, probability should be assigned according to attendee’s preference. Additionally, some destinations setting only reflect general directions not specific spots because our simulation goal is to help pedestrians get out of campus.

To simplify the behavior of pedestrians, we keep their walking speed constant as 1.3 meters per second. Even though, this simulation is not very accurate as different people have distinct walking speed and the speed varies as location and time changes, it could capture the behavior of pedestrians as it conforms to the total egression time from the stadium. What’s more, we assume all pedestrian follow the same rules of travel, which means they will not engage in any form of travel other than by foot. This simplification also includes that no pedestrian is affected by endogenous factors such as emotional or radical behaviors or by exogenous factors like raining, wind or chaos.

First, we suppose that the only pedestrians in the model are ones that have exited from the stadium and that the stadium is at full capacity. Besides, Assumption needs to be made that each person has decided a destination, which will not be changed once they leave the stadium. In addition, for people on their way to destination, the path they are going to cover is an optimal path and this path will not change as people approaches their destination. Thus we assume that pedestrian do not make path decisions for safety, sightseeing, or other reasons. Along the path, crosswalks are considered walking terrain and we assume crossing guard allow pedestrians walk instantaneously if they get there. Furthermore, for purpose of circumventing insignificant details, no congestion at the destination is assumed, which means there is an immediate exit of the system when the pedestrian arrives at the final destination and no wait or queuing phenomenon occurs. This might be unrealistic in several cases because there always existing a waiting time for people to board a MARTA train or bus. But since people have already been distributed so sparsely that they will not affect the march of the following pedestrian and destinations are selected far away from stadium so as to it is unrelated to the system, this assumption is valid and helpful for us to simulate.

The output of the model will be a random variable corresponding to the elapse time it takes for all pedestrians to exit Georgia Tech's campus. In terms of the goal of minimizing the total retreat time, we will be able to establish some confidence interval based on the retreat time found from the simulation. Using the retreat time from several simulations we can determine within what certainty our sample mean coincides with the true simulated mean. This is the parameter that we are trying to optimize to achieve the project goal of evacuating pedestrians as quickly as possible.

IV. Simulation Descriptions

The program has three main parts: File I/O, Simulation Engine, and Data Structure part.

First, the File I/O part takes charge of reading the input file. There is a fileReader method in the FileIO class, which used to read the input "new_map.png" file and generate the corresponding map object. Also, a class called Plot_map is created to help visualize the pedestrian movements and generate corresponding results.

Second, The Data Structure part includes all the data structure/classes the simulation program required to use. It mainly contains two class Roadmap and Location. For the Roadmap class, it stores the map information in the program. The map class contains several variables. For example, the container is used to save the map information by cell. Location class is the parent class for several classes such as Entry and Pedestrian. Entries, a list of entry coordinates, which is used to store the information about entry. Same story for Exits. For the Pedestrian class, it stores the information about each pedestrian and it has the variable such as the pedestrian location and his/her destination.

In addition to Roadmap, there are some other classes used for better classification of the entities on the map. The parent class is called Cell, and several classes is inherited from it. They are RoadCell, BlockCell and ExitCell. Cell is the parent cell class which presents each cell in the map. And other classes like RoadCell and BlockCell are subclasses of Cell and these classes have specific functions. For each cell it has a property about whether this cell is occupied by a pedestrian. And for the Block Cell, it is always occupied so that no pedestrian can get to the block cell.

Third, the simulation engine is implemented in Simulator class. It contains all the functions needed for simulation process. There are two major functions: initialize, update_map. They are used to initialize new pedestrians to simulation system and update pedestrians positions on the map.

A. *Initialize function:*

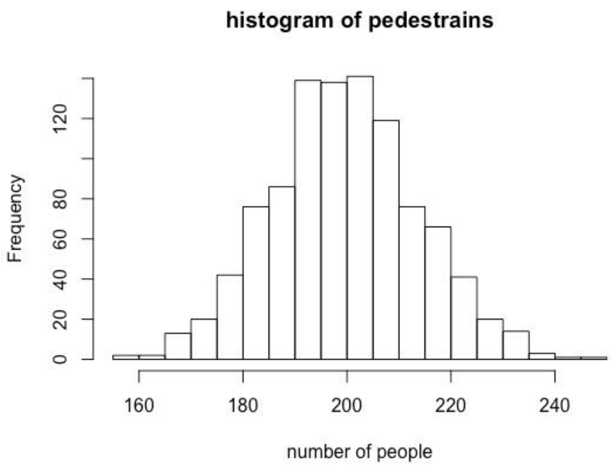
Function initialize generates pedestrians at the stadium gates (entries in our program). It gives the pedestrian a random enter position among entries and set a stochastic destination for him. We assume that number of pedestrians leaving stadium at each timestamp is relevant to time. Then, we come up with polynomial function to depict this relation. We refer to a study of evacuation[3] and decide to use Poisson distribution to simulate the number of pedestrians generated at each gate. It satisfies following conditions:

1) The integration of this function should approximately match the capacity of stadium used in the model.
2) The maximum value in the function should not exceed the limit of gate numbers.
3) The end point in the function plot should match the end of evacuation process.

Coefficients of the Poisson Distribution function is decided by the conditions above. In addition, number of pedestrians coming out of the gates is set to be equal to the number of total gates, which means all the gates are occupied during the egress process. For our simulations, we set the parameters of poisson distribution based on the number of total gates. And parameter changes as the simulation process goes. At the very first stage, the parameter is equal to number of total gates. And as time goes by, it changes to half of the gate number. Later, it reduces more to one third of the gate number.

Figure I comes from a poisson distribution with mean 200. It looks like normal distribution but we will stick with poisson because poisson distribution is mostly used for count variables. The number of pedestrians is a kind of count data, then the poisson distribution will be the choice for our random number generator.

FIGURE I. Poisson Choice for Pedestrian Distribution



B. *Update_map function:*

The update function is based on static floor. The moving direction for the pedestrian is decided by the shortest path to the exit. Before the simulation begin, the program will generate several 2D arrays. The cell in each array is used to save the shortest distance from each cell to the destination. The shortest path array is able to lead the pedestrian to the right place. In addition, to make the simulation more similar to the reality, the pedestrian will not always go to the right neighbor which is with shorter distance to the destination. Therefore, the update function will generate a random number, based on which the pedestrian can choose a stochastic movement for the next step.

V. Simulation Results

From the simulation graphs below, we can have an idea how our simulation model work. Red dots represents pedestrians. Blue dots represent the entrance of stadium, where is the entry of our pedestrians. They leave from the entrance to exists near the border of the simulation map.

Our simulation is conducted based on following parameter settings: the total length of simulation run is 15,000 timestamp and the total number of pedestrians evacuated from stadium is 15,000. The simulation will be finished within 15,000 timestamp length.

The exits are read from map based on the color red. Our simulation takes about 10 minutes to get all the information ready from the map (The map provided contains hundreds of exit pixels). After the "set-up" wait, the simulation will begin and corresponding graphs and status will be printed on screen. Our simulation is programed in Python 2.7.13. This simulation takes about 1 hour to finish. The computer we use to conduct the simulation is a Macbook Pro, with 2.6 GHz Intel Core i5 and 8GB 160 MHz DDR3.

FIGURE II. Pedestrian Distribution at Timestamp 200

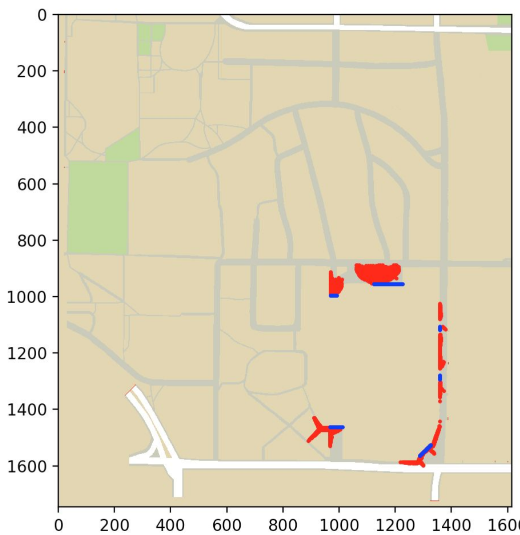


Figure II shows the distribution of pedestrians at timestamp of 200. We can easily see that there are six exit gates around the stadium for people to evacuate. People are clustered near the entrance because they the time is so short that the distance of path from the standing spot to starting point is tiny and invisible. However, the pedestrians' direction can be easily distinguished.

FIGURE III. Pedestrian Distribution at Timestamp 500

Figure III shows the distribution of pedestrians at timestamp of 500. At this time point, more pedestrians generated and the pedestrians generated before have moved around to get to its exit.

FIGURE IV. Pedestrian Distribution at Timestamp 3,000

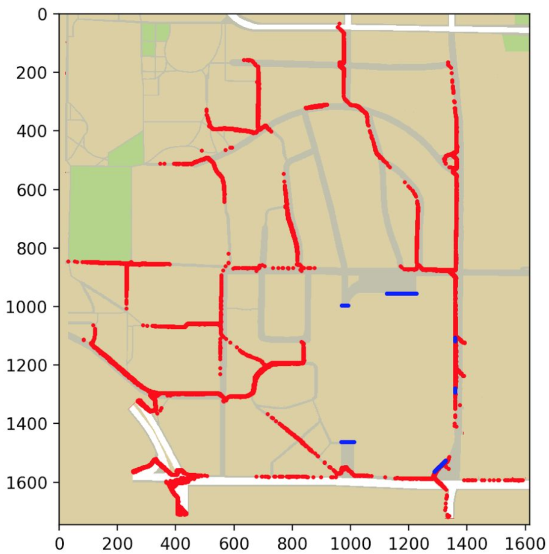


FIGURE V. Pedestrian Distribution at Timestamp 5,000

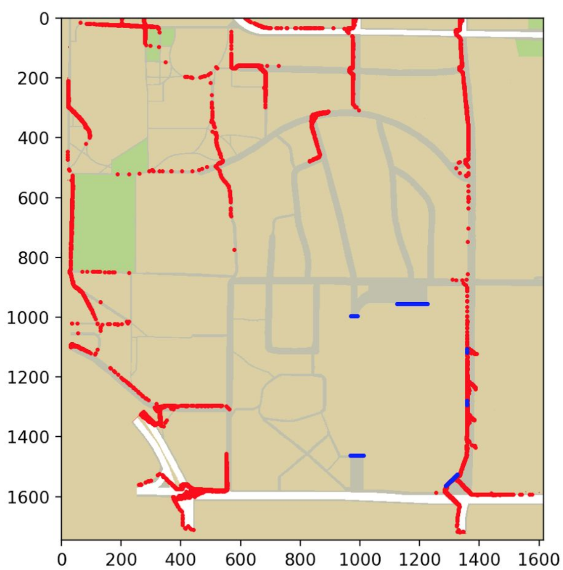


Figure V shows the distribution of pedestrians at timestamp of 5,000. We can see that as time goes by, pedestrians spread out more and most of them are heading to their destinations.

And Figure VI shows the distribution at time stamp 8,000 , which indicates that number of pedestrians on the roads is at pretty low level. Most of pedestrians have evacuated from the GT campus at this time.

FIGURE VI. Pedestrian Distribution at Timestamp 8,000

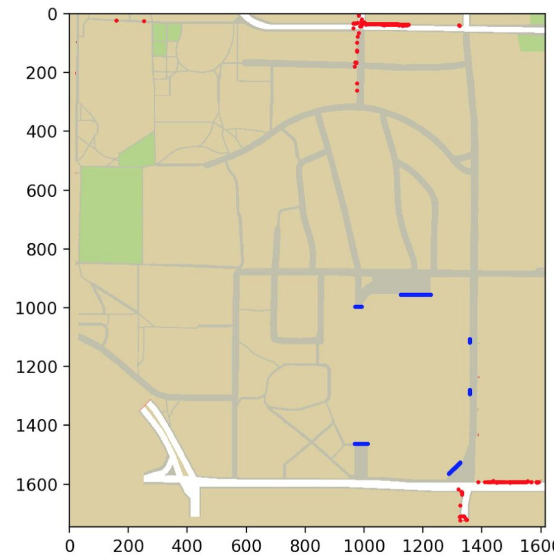


FIGURE VII. Pedestrian Distribution at Timestamp 13,000

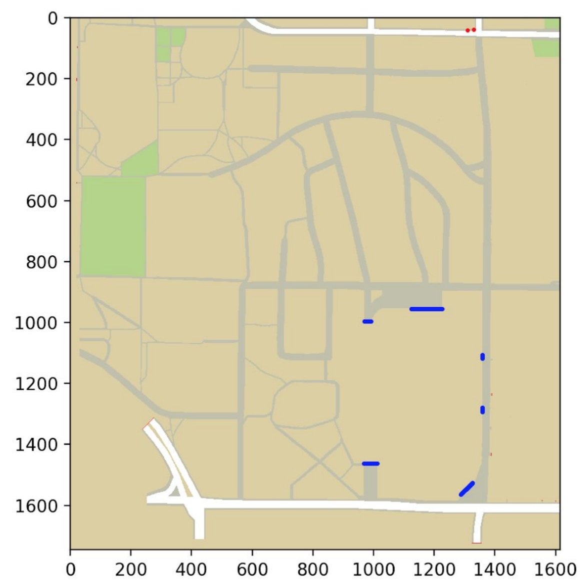


Figure VII shows the distribution of pedestrians at timestamp of 13,000. At this time stamp, all pedestrians find their exist and go out of the simulation system except 2 persons at the top-right corner. They exit after this timestamp. And from this timestamp on, all pedestrians get out of the simulation system.

The series of figures show the evacuation process simulated by our program, the results match our expectation and the relationship between time stamp and number of pedestrians on the map share some properties with observations in others' studies.[5]

VI. Conclusion

In all, our model simulated the whole process which ended at timestamp 15,000. As expected, the evacuation takes advantage of every road cell, and all 15,000 pedestrians from the stadium successfully evacuated. The accuracy of pedestrians movement simulation is improved by the automata model. The results are reasonable compared to ordinary evacuation observation. Therefore, the model in the reported can be applied to other topographs holding large community event.

Our model and method have univariate usage as long as the input map is recorded using the color as the one we use. In the future, other statistical models like normal distribution, exponential distribution can be applied to simulate pedestrians movement and the outcome rate at the entries.

References

[1] Ribeiro, João, et al. "Towards a serious games evacuation simulator." arXiv preprint arXiv:1303.3827 (2013).

[2] Stanley, H. E., et al. "Physica A: Statistical Mechanics and its Applications." Proceedings of the 1996 Workshop on Current Problems in Complex Fluids. 1997

[3] C. Burstedde, K. Klauck, A. Schadschneider, J. ZittartzSimulation of pedestrian dynamics using a 2-dimensional cellular automaton Accepted for publication in Physica A. 22 pages, 10 figures (2001)

[4] Wishart, David S., et al. "Dynamic cellular automata: an alternative approach to cellular simulation." In silico biology 5.2 (2005): 139-161.

[5] Kirchner, Ansgar, and Andreas Schadschneider. "Simulation of evacuation processes using a bionics-inspired cellular automaton model for pedestrian dynamics." Physica A: statistical mechanics and its applications 312.1 (2002): 260-276.

[6] https://en.wikipedia.org/wiki/Poisson_distribution