MODELING OF PEDESTRIAN TRAFFIC AROUND BOBBY DODD STADIUM

CSE 6730 PROJECT 1 REPORT
SPRING 2016

Fred Hohman Devavret Makkar Nanditha Rajamani

CONTENTS

1 PROBLEM STATEMENT

College football is often characterized by passionate fans, intense rivalries, and sold-out Saturday games. Georgia Tech, a team with much history, is no different. Saturdays in Atlanta during the college football season are packed with fans: students, alumni, and miscellaneous visitors. Regardless if Georgia Tech wins or loses, a sold-out Bobby Dodd stadium creates a large amount of hustle and bustle on nearby sidewalks and streets.

This project aims to create a model to simulate the foot traffic after the conclusion of a Georgia Tech football game. It should be noted that we will be only be modeling *foot* traffic, so no vehicles will be considered. Furthermore we will only be modeling the flow of people outside the stadiums walls, but these restrictions are further detailed in the conceptual model below.

Our goal is evacuate the as efficiently as possible, i.e. we want to minimize the total amount of time required to move people out of the immediate area.

2 Literature Review

Background

The book “Pedestrian and Evacuation Dynamics" contains a vast amount of research topics and general information relating to efficient flow of people. In particular, the importance is described in the following quote: “Proper management of evacuation processes is one of the basic requirements within life safety concepts, and it helps to prevent critical situations from getting out of control" [2].

This book described the typical cellular automata (CA) based behavior model as pedestrians' walking paths represented by the cells of a 2D lattice. As learned in class and emphasized by this specific text, the pedestrians in a CA model update simultaneously in discrete time steps. Moreover, all pedestrians follow some local rules defined exclusively on a pedestrians previous state and its neighboring cells. During each time step a particular cell considers each neighboring cell as a potential next destination assuming no other pedestrian, obstacle, or collisions would interfere. Depending on the specific application these rules are defined to govern the general flow of pedestrians. Our project will need to consider both physical and social factors when creating our rules.

As a note, when we say neighborhood we use the notion of a Moore neighborhood in which the number of cells n in a neighborhood of cell x (where x is excluded) is determined by the expression

$$(2r+1)^2 - 1.$$

When we take $r = 1$, we see our result says a neighborhood contains 8 cells. We will adhere to the $r = 1$ for our project, although if needed we can use an extension of the Moore neighborhood in which we count the central cell, simply adding 1 extra pedestrian to our total in order to give the option of a pedestrian the option of not moving, i.e. standing still [2].

Another source of information comes from a Ph.D. student's dissertation titled “Modeling Behavior in Vehicular and Pedestrian Traffic Flow," in which the author, Markowski, once again describes the mechanics of cellular automata. Markowski describes a study in which a cellular automaton model and its rules were applied to both a highway traffic model and a pedestrian environment model separately, and unsurprisingly emphasizes how important the cellular automaton rules really are. He goes on to explain since all (or many) cells are controlled by the same set of rules, the smallest change in the rule-set can have “wide ranging effects on the ultimate system organization" [3], once again enforcing the importance of clearly defining and justifying our project's rule-set.

TOY EXAMPLE

Markowski explains the rules given for the vehicular highway traffic flow, but since our project will simulate pedestrian flow the details are not relevant; however, the author gives some background on a simple set of rules coined by John Conway titled the “Game of Life." The rules are as follows. Each cell is in one of two states: *dead* or *alive*. For each time step, each cell makes the following update:

1. If alive and has less than 2 living neighbors, cell dies (lonely).
2. If alive and has more than 3 living neighbors, cell dies (overcrowded).
3. If alive and has 2 or 3 live neighbors, no change (healthy).
4. If dead and has 3 neighbors, cell comes to life (birth).

Since it is rather difficult to predict how a system such as the previous will evolve over time, this simple example emphasizes how powerful cellular automata is.

Markowski discusses some challenges of pedestrian flow compared to vehicular flow by stating that pedestrian traffic modeling is more difficult since vehicular traffic is primarily organized in lines, obeys traffic laws, and communicates using lights and signals. Pedestrians on the other hand, ignoring cultural norms and physical barriers, are free to roam where they like, at whatever pace, and communicate using body language as opposed to discrete signals (much harder to quantify). Markowski concludes by describing physics-based models as a potential for improvement, but that is beyond the scope of this class project [3].

CROWD CONSIDERATION

Crowd simulation models can be used to predict crowd efficiency and performance issues in the design of buildings and other public facilities. An important application of such models is the estimation of evacuation time, which will eventually help develop safe and efficient evacuation plans in case of an emergency situation.

Cellular automata, developed by John Van Neumann and Stanislaw Ulam, is extensively used for modeling entities in a “cell", which live on a grid, have a state and are surrounded by other cells (neighbors). Features such as simplicity, modularity, low computational complexity, make cellular automata an extensively used modeling paradigm from simulating microorganisms in biology to a large body of people in crowds[4] [5].

Crowds can be considered as collective movements of individual pedestrians. Therefore, there are a plethora of methods in which they can be modeled. In general, they can be divided into two categories: Microscopic and Macroscopic models. Macroscopic models focus on overall pedestrian movement situations and view pedestrians as flow. Lattice gas model, fluid-dynamic models and network models commonly use the macroscopic approach [4].

Microscopic methods simulate pedestrians as individuals and describe them with individualized behavior with greater details and hence require greater computational complexity. Models which follow this methods are again divided into several categories such as physical based models, rule based multi agent based model, and cellular automata.

Though each model has it's merits and demerits, we choose to use a cellular automata to simulate pedestrians as it is most optimal to satisfy the goals of our project.

Qualitative Data Collection

Pedestrian behavior has been studied for more than four decades now in both normal and panic situations as it provides great insight into development of evacuation plans. Most of the data we have summarized here, was determined through literature survey, [6] has used time lapse films, direct observations and photographs to make observations and conclusions on the relative speed of walking, the choice of which route to take and distance between two pedestrians and obstacles:

1. Pedestrians are adverse to moving opposite to the desired walking direction, or taking detours despite the direct route being crowded. There is evidence that pedestrians tend to choose the fastest route, which may not be identical to the shortest route. The end goal seems to be minimization of the effort to reach their destination.

2. The speed which with pedestrians walk correspond to the least energy consuming walking speed. These represent a Gaussian distribution with a mean value of 1.34 m/s and a standard deviation of about 0.26 m/s. However, it must be noted that these speeds are subjective and change, albeit to a small degree, with change in sex, age, time of the day, destination, prior and current emotional state, etc [10].

3. The distance between pedestrians and between obstacles/borders decreases as the pedestrian increases the speed of motion(i.e., is in a hurry) and it also decreases with growing pedestrian density.

4. The average sidewalk width as specified by the American Disabilities Association is considered to be the standard, and measure 5 ft in length. However, as discerned from using the android software developed by Georgia Institute of technology provides a mean value of 9ft, with a standard deviation of 4.035 ft[10].

Formalization

The use of two-dimensional cellular automaton have been demonstrated to be effective for simulation of pedestrian traffic [1]. The pedestrians are modeled as particles that can take a position in a grid. No two particles can share a grid position. At every time step, all pedestrians are moved to a neighboring cell based on the floor fields and collisions are resolved by the relative probabilities with which the pedestrian chose that cell. Two fields are used: static and dynamic. Static fields do not evolve with time and are used to specify regions of space which are more attractive eg. exits. Dynamic fields are modified by pedestrians and are increased when a pedestrian walks over a place. They decay with time and are only regenerated when another pedestrian steps onto that cell. These are used to model the behavior of pedestrians' likelihood to follow others before them. The transition probability in any direction is given by

$$p_{ij} = NM_{ij}D_{ij}S_{ij}(1 - n_{ij}) \tag{2.1}$$

Here n_{ij} is the occupation number of the target cell in direction (i, j), i.e., $n_{ij} = 0$ for an empty cell and $n_{ij} = 1$ for an occupied cell. Therefore transitions to occupied cells are forbidden. N is a normalization factor to ensure $p_{ij} = 1$ where the sum is over the nine possible target cells. D is the dynamic floor field modified by presence of pedestrians and S is static field that remains constant for a cell throughout the simulation.

3 Conceptual Model

As highlighted previously, the project involves developing a simulation model of people leaving after a football match that occurs at Bobby Dodd stadium with the main objective of minimizing the evacuation time.

Initial Physical Facts In planning the conceptual model, our group identified important physical parameters that must be taken into account in order to conduct a representative simulation. Often times said parameters are fact but others must be approximated and assumed. Listed below are these parameters and assumptions:

- Bobby Dodd Stadium capacity: 55,000 people, stadium is at near capacity.
- Average speed of walking is represented by a Gaussian distribution, and can be approximated as 1.34 m/s (3.0 mph) with a standard deviation of 0.26 m/s (0.58 mph).
- Average width of the sidewalk, as gathered from the "sidewalk app" developed at Georgia Institute of Technology is 9.0 ft with a standard deviation of 4.035 ft.
- Each cell represents an individual and no vehicles are considered. All pedestrians obey traffic signals and stay on the sidewalks.
- We will decide which roads are closed off in order to simulate as accurate as possible where people are able to move given the circumstances a football game has the road network.
- The five exit points we define are in the direction of the North Ave MARTA station, the Clough Undergraduate Learning Center (CULC), the Klaus Center for Advanced Computing, Tech Square, and the North Ave Parking Deck.

The model is based on 2-Dimensional grid over the map of Bobby Dodd stadium. The cells are square and one cell is occupied by one individual. The cells that lie outside the sidewalks are rendered invalid, thereby simplifying our computation. In order to model barriers, walls, buildings and other obstacles will be rendered ineligible by our force fields. More barriers can be defined as we finalize our segmented map of Bobby Dodd.

GRID LAYOUT To obtain an initial map of our considered area, we use Google maps. The issue of scaling is dealt with by assuming that in a crowd, a single individual takes up approximately one square foot of area on the ground. This leads to an assumption that 1 pixel is approximately equal to 1ft^2. Thus, each person occupies one square foot of area when digitally represented. The browser window displaying the map is then printed (screen shot) to take advantage of the fact that computer screens are already discretized by pixels. The image is segmented to transform it into a matrix of different pixel values in order to obtain a foundation of our model. This segmentation will separate eligible form ineligible movement spots for evacuating people. This process will be elaborated on in the implementation section.

INPUT PARAMETER ANALYSIS Recall Alexadra Frackleton used time lapse films, direct observations, and photographs to make observations and conclusions on the relative speed of humans walking, the choice of which route to take, and distance between two pedestrians and obstacles [6]. Using what we learned in our literature survey, we conducted our own observational experiment to gather some real world data. From the back of a mid-sized classroom, we filmed the end of class while people exited out of two doors in the front of the room and recorded at which times a person left. The histogram shows that as soon as the class (or in general, an event) ends, there is an initial surge of people leaving the room, but at some point the amount decreases quickly until only a few people trickle out towards the end. We used a Gamma distribution to represent this curve and scaled it up so that the area under the curve was equal to the total number of people in our simulation: 55,000.

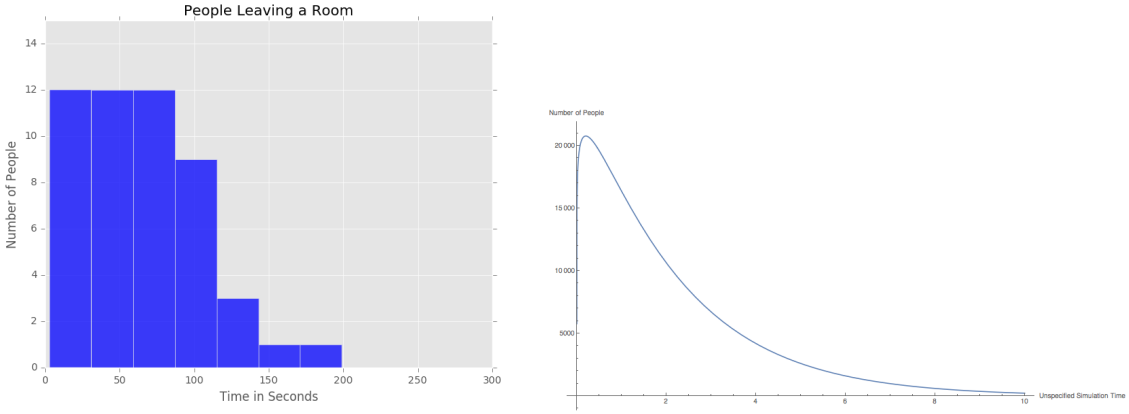


Figure 3.1: Our observed histogram of people exiting a classroom and the theoretical interpolated function of fans exiting Bobby Dodd.

Mathematically we can represent this as the following equation:

$$\Gamma(k,\theta) \equiv \text{Gamma}(k,\theta) \implies \Gamma(1.1,2) = 0.490373 e^{\frac{-x}{2}} x^{0.1}, \text{ for } x > 0,$$

where our property is met:

$$55000 \int_0^\infty \Gamma(1.1,2)\,dx = (1) \times 55000 = 55000.$$

Using the percentages from our observed experiment and knowledge of what the curve looks like as people evacuate a room, we constructed a new function in order to decide how to introduce, i.e. spawn, people into our simulation. The function inputs the number of people already spawned in the simulation and outputs the percentage of a region to spawn people accordingly. The implementation section will describe in more detail how we went about choosing the regions to spawn people, but the relevant information is that given a spawning region, for every time step we want to introduce a certain number of people to the simulation. Our regions are represented as long, thin rectangles. We decided the maximum number of people to spawn for a given time step was 50%. So the function we constructed should satisfy the following evaluations:

$$f(0) = 0.5 \implies 50\%, \quad f(55000) = \text{some small number that is not quite } 0$$

With this in mind, we present an simplified input distribution modeled by exponential decay that determines how many people to spawn per time step.

$$f(x) = \frac{1}{2} e^{-0.00005x}$$

Notice also that $f(55000) = 0.0319639 \implies \sim 3.2\%$, which is a small number like we wanted.

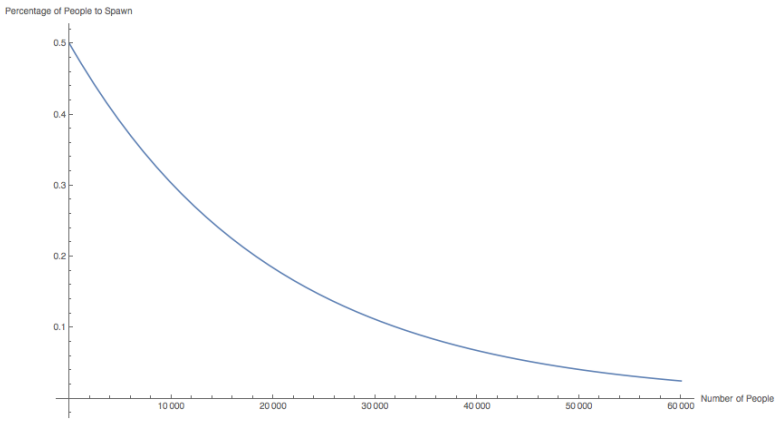


Figure 3.2: Our spawning function for determining how many people to introduce per time-step.

Notice the curve is qualitatively similar to that of the theoretical Gamma distribution described above. Both the Gamma distribution and our function are based off of experiential data, so we consider both to be approximations. We think our function accurately represents physical reality, and we use it in our software implementation.

MODEL DESCRIPTION At every time step a pedestrian will walk to a neighboring legal cell (8 choices: up, down, left, right, and all four diagonals). The movement per time-step is restricted to one cell because on an average, all persons in a crowd are moving at the same pace. In situations like exiting a stadium, people do not have the freedom to walk at a faster pace as their path is already packed with other people moving at a steady pace.

To approach the end objective of minimizing the evacuation time, we implemented the Breadth-First Search (BFS) algorithm as the preferred traversal method of people. Obstacles such as walls and buildings are placed at realistic positions and people dynamically move around them to reach their destination. Once every person has reached his or her assigned exit the simulation ends.

USE OF FLOOR FIELDS The next aspect of the model is the definition and use of three different "flood fields," namely: static fields, dynamic fields, and anticipation fields. The static field is defined to remain constant with time and serves as an "attractor" for certain points in the map such as exits. The dynamic field, on the other hand, decays with time and is modified by people. It provides the cells with memory to simulate people moving in clusters and groups. In keeping with the rules of our model, where two people cannot share a cell there is a need to resolve conflicts. This is done by modeling an anticipation field that is described in detail below.

STATIC FIELD Defining the static field has been studied by Huang at al. [9] where they start with $k = 1$ and start labeling all grid cells in the immediate one-hop neighborhood of the exit (door/destination of pedestrians) with k. Then they increment k by 1 and repeat this step for the cells at a two-hop neighborhood of the exit. Thus they create a grid of static field with the value highest for cells at a maximum distance from the exit. Pedestrians would then want to go from a higher number of static field to a lower number, ending up at the exit which has the lowest number, i.e 0.

DYNAMIC FIELD Diffusion and decay of dynamic fields are defined by [7] as below

$$D_{ij}(t+1) = (1-\delta)\left[D_{ij}(t) + \frac{\alpha}{4}\Delta D_{ij}(t)\right]$$

where

$$\begin{aligned}\Delta D_{ij}(t) = {} & D_{i,j+1}(t) + D_{i,j-1}(t) \\ & + D_{i+1,j}(t) + D_{i-1,j}(t) - 4D_{ij}(t)\end{aligned}$$

and D is the dynamic field floor, $\delta = 0.1$, $\alpha = 0.3$.

ANTICIPATION FIELD Y. Suma et al. [8] developed an additional factor called the anticipation field. This corresponds to pedestrians ability to avoid collisions with oncoming traffic by anticipating where they are headed and avoiding stepping into that path.

4 SOFTWARE DESCRIPTION

Our software is divided into 3 types of components and separates the simulation engine from of people movement rules and heuristics:

- Classes
- Libraries
- Simulation Engine

CLASSES We currently have two classes: Human and Floor. Human class has members position, destination and field. Field here stores the human's current walking direction.

For Floor class, we store the size of the floor field, position of human beings on the floor, various static fields corresponding to the exits, dynamic fields left by humans walking in various directions, and a 2D array to store whether the corresponding location of the floor is a wall or not. This class also has methods that are called by the simulation engine to grow static field from a point using BFS, to calculate dynamic field (diffusion and decay) and to calculate anticipation field for each human on the floor.

LIBRARIES As of now, we have only one library which writes bitmap images. It has two overloaded functions to write a gray-scale or a color image. The gray-scale writing function is used to write images of floor fields as a heat map. The color bitmap writer takes the positions of the human beings as input and marks their positions using different colors based on the category of humans. Here category means the destination they want to go to. So all humans going to MARTA will be colored the same color.

SIMULATION ENGINE The simulation engine uses the classes to simulate the movement of human beings based on the rules described in the conceptual model. The Simulate() function is described as:

Data: Floor object, vector of humans
while *(current time < max time)* **do**
    Walk all humans;
    Resolve conflicts;
    Remove exited people from simulation;
    Increment time;
    if *time % 100 = 0* **then**
        Write bitmap;
    end
end

Algorithm 1: Simulate function

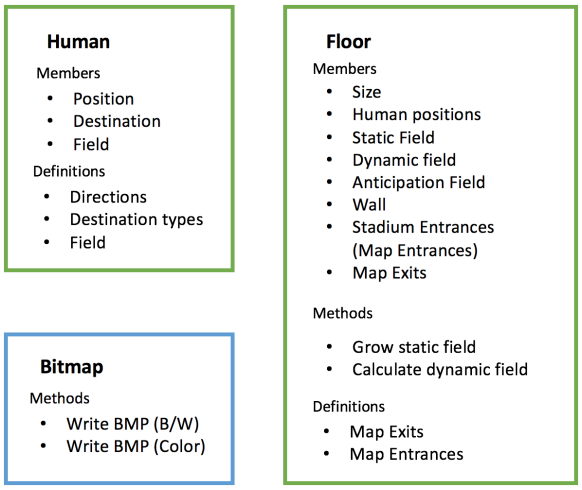


Figure 4.1: Our simulation algorithm.

5 Implementation

This section contains a handful of key components and assumptions of our simulation model.

Random Number Generator As mentioned in the project prompt, we implemented our own random number generator. We chose to use a multiplicative linear congruential generator, typically associated with Lehmer's algorithm. The general formula is

$$X_{k+1} = a \cdot X_k \mod m,$$

where the modulus m is a prime number or a power of a prime number, the multiplier a is an element of "high multiplicative order modulo m", and the specified seed X_0 is coprime to n. We scale the generator so that we only output numbers between 0 and 1. Specific values of a and m have been published that produce the best results, so instead of choosing our own numbers we select the popular combination $a = 16807$ and $m = 2147483647$ [11]. We also account for overflow on certain systems. A benefit from this type of RNG is that it is fast and easy to implement. Furthermore, we can pick the seed in order to achieve reproducible randomness for testing our simulation.

Map Generation and Entrance/Exit Classification We identified two possible approaches to identifying map and grid generation: either select the areas people are allowed to traverse or not allowed to traverse. Both have pros and cons, however we decided to select the paths by which people are allowed to walk. This enables us to direct the flow of movement easier, especially if our goal is to evacuate people as fast possible. We believe this is more physically representative too, since moments after a game concludes roads are closed and police forces direct fans safely in specific directions. If we want to select which roads, sidewalks, and paths people are allowed to take, we needed to make sure our method was easily scalable so that if a new path looks like a reasonable path, it can be easily added to the simulation.

We start by navigating to Google Maps and zooming in so that the scale is exactly 100ft for 100 pixels. This gives us a pixel area of 1 pixel = $1ft^2$. Once this scale is achieved we stake various screen-shots of the areas immediately surround Bobby Dodd. We then splice the images together so create a large map. Once completely, we decide on the entrances to our model (the exit gates of the stadium) and the exits of our model. We used an image from the official Georgia Tech Athletic website to obtain the following figure [12].

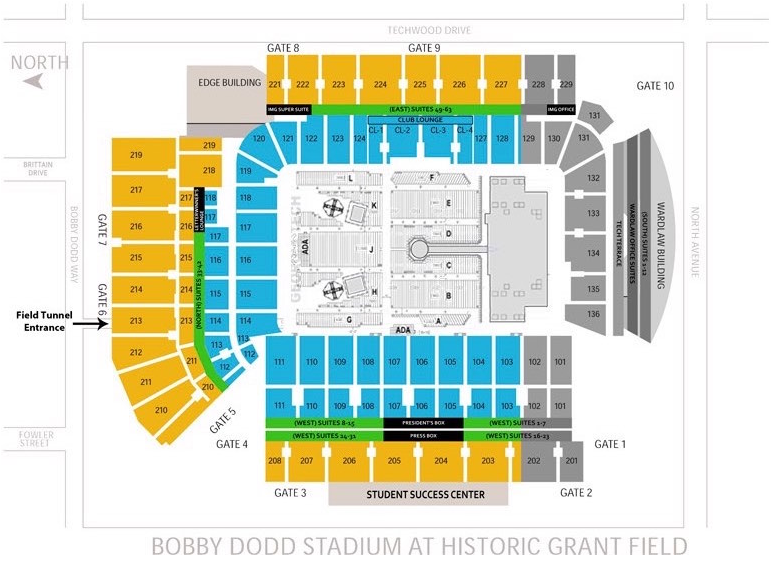


Figure 5.1: The 10 identified Bobby Dodd Stadium Gate Exits.

Once we knew the gate locations, we obtained the corresponding pixel coordinates on our map. We then identified the five most common simulation exits identified previously. In an effort to keep our simulation time down to perform the necessary testing due to time constraints, we do not include the actual exits in the map; we simply say once a person has reached the edge of the map in the direction of their exit, then they will not have any trouble completing the distance. We believe this is physically representative since most crowd commotion occurs directly outside of the stadium, but once people start walking where they want to go (car, specific building, etc.) then lanes and paths form uninhibited. For further evidence that is is physically representative, we know fans are given specified seats and must enter the stadium at certain locations regardless of how they arrived toe the stadium. So we expect a great deal of movement directly surrounding the stadium but more straight lanes approximately a block outside the stadium.

Now that we have identified our entrances and exits for the fans in our model, we now needed to define the paths they can walk. Instead of guessing the sidewalk length on every road, since we have scaled our map accordingly, we were able to load our map into a photo editing program and trace over the paths we allowed to people to walk. These included the specific roads that are closed during game-day, the sidewalks on roads that are not closed, as well as foot traffic only paths on campus. Once we trace the exact path we allow people to go, we wrote a script to segment the image into 0s and 1s, where 0s represent the paths

and 1s represent boundaries. This is easily imported into our software to set the boundaries for the simulation. We do not account for cross walks in our model, as many of the paths we chose are foot traffic only and closed roads. On the few intersections we do model we make the assumption from personal experience that given the chaos of game-day, people will largely dominate cross walks and cars will be consistently yielding.

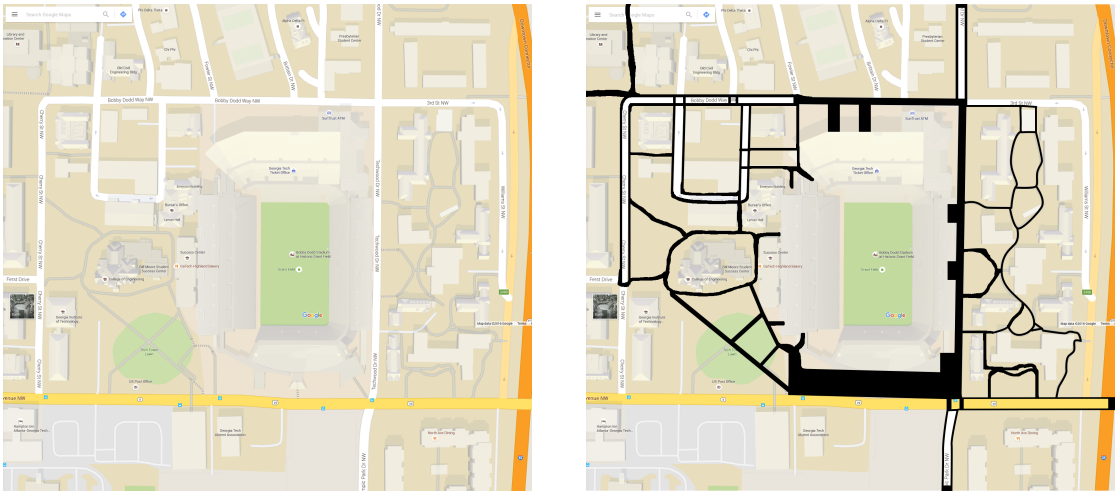


Figure 5.2: Our identified map and the allowed paths.

Most importantly, the process of tracing exactly the width of sidewalks and paths can easily be extended; we simply add the paths in our image program and rerun our script top generate a new map of 0s and 1s. We think this scalable map generation is flexible and quick enough to test future maps if given time.

INPUT, MOVEMENT, AND EXITING IMPLEMENTATION DETAILS We spawn people at each of the ten gates/entrances randomly according to a uniform distribution, that is, if we call a random number x, if $0 < x < 0.2$ this person is assigned to exit 1, and if $0.2 < x < 0.4$ then this person is assigned to exit 2, as so on. We can seed our random number generator to give us different movement results. We used a storage list of people so that a person could be added and removed upon spawning and exiting the simulation. Once we were able to include an arbitrary amount of people to the model, we scaled this up to include 55,000 fans: the capacity of Bobby Dodd.

We move people using static fields: a type of force field that pushes people towards one central point. We are able to specify the exact coordinates of an exit in order to calculate a static field that pushes all humans towards said exit. This is done by using a breadth first search algorithm to calculate the minimum distance from each exit to every other eligible spot on our grid. So for five exits, we have five separate static fields, and each person follows a static field based off of their exit assignment.

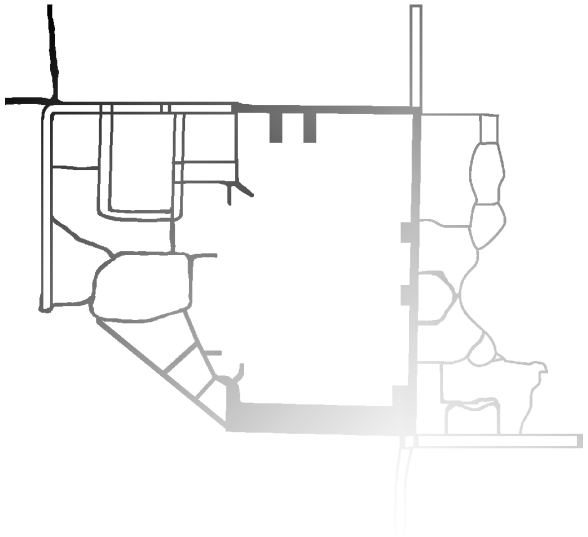


Figure 5.3: The static field for the North Ave Parking Deck.

Next we added a collision detection and resolution procedure so that given a certain time step, if two people want to move to the same spot, one person will move there and the other will stand still. We decide who takes the spot by performing an array shuffle on the people trying to enter the same spot and randomly choosing from that list, introducing another element of randomness to our simulation.

For exit criteria, we specified regions near the borders of the map that remove a person from the simulation once said person's coordinates are inside the box. As already stated above, we assume once a person has reached a distance this far form the stadium, he or she will be able to continue towards the actual exit location without halt.

6 VERIFICATION

As we built our model and included more features, we use an iterative design process that helped us tweak our conceptual model in order to better align with our current code implementation. In this section we will highlight a few key features we have successfully implemented as well as provide visualization images to both verify and validate our model.

It is also worth noting that our software contains two primary functions: Simulate() and SimulateTest(). Both functions use the same rules, however SimulateTest() runs the simulation on a small 10×10 grid so we can verify every rule and perform tests to ensure our code is functioning as intended.

VERIFICATION TEST I: STATIC FIELD AND MULTIPLE PEOPLE Our first goal for our simulation model was manually programming a single human to walk on a grid. Once this initial step was completed, we set out to implement the various features of our model. This first example tests three primary features: multiple people walking on a grid, hard exclusion (i.e. collision resolution), and static force fields.

Since we will spawn people inside of Bobby Dodd stadium based on a statistical distribution, we needed a data structure and method to hold all of the people in our simulation. We were able to implement a storage list of people so that a person can be added and removed upon spawning and exiting the simulation. Once we were able to include multiple people, we can scale this up to include as many people as we (or our computers!) can handle. Next we added a collision detection and resolution rule so that given a certain time step, if two people want to move to the same spot, one person will move there and the other will stand still. Finally, we implement static fields: a type of force field that pushes people towards one central point. We are able to specify the exact coordinates of an exit in order to calculate a static field that pushes all humans towards said exit. This is done by calculating the minimum distance from each exit to every other eligible spot on our grid.

Our first example output is visualized below. As stated earlier, we visualize the exit as a white square at coordinate position $(2,1)$ (assuming a zero-indexed matrix style origin placement at the top left), and each darker spot represented the minimum distance to the exit. The blue spots are each one human.

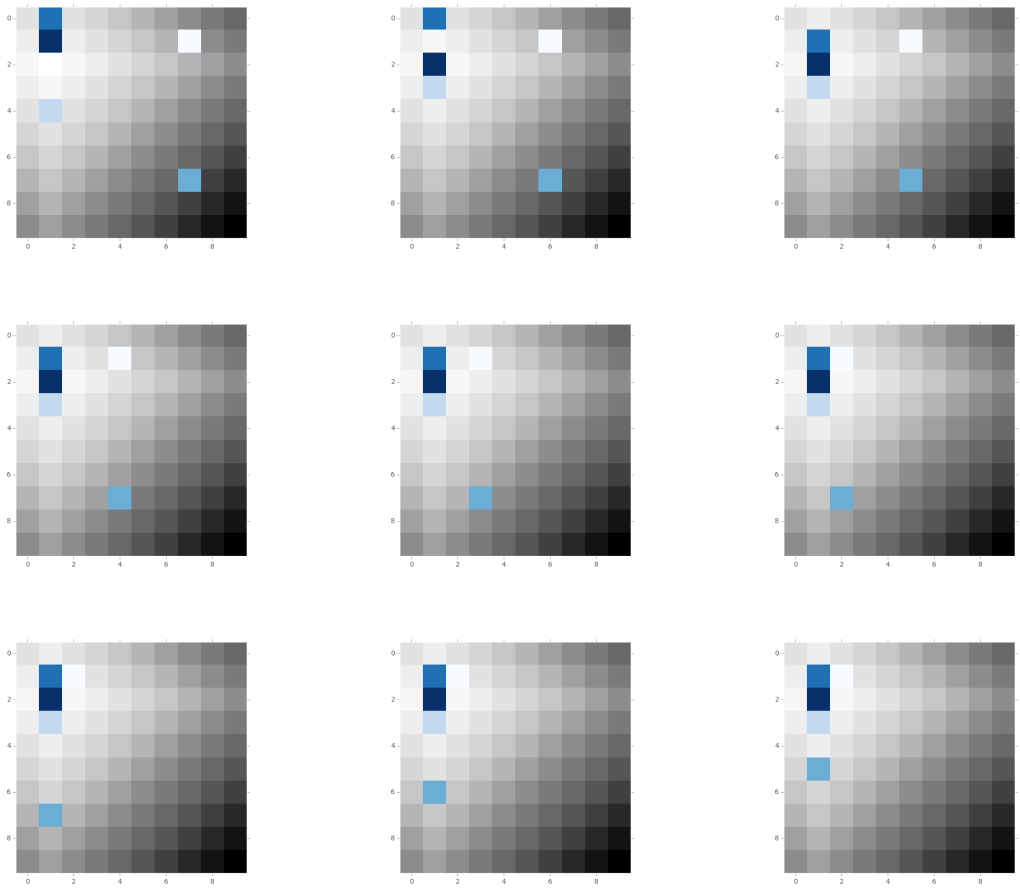
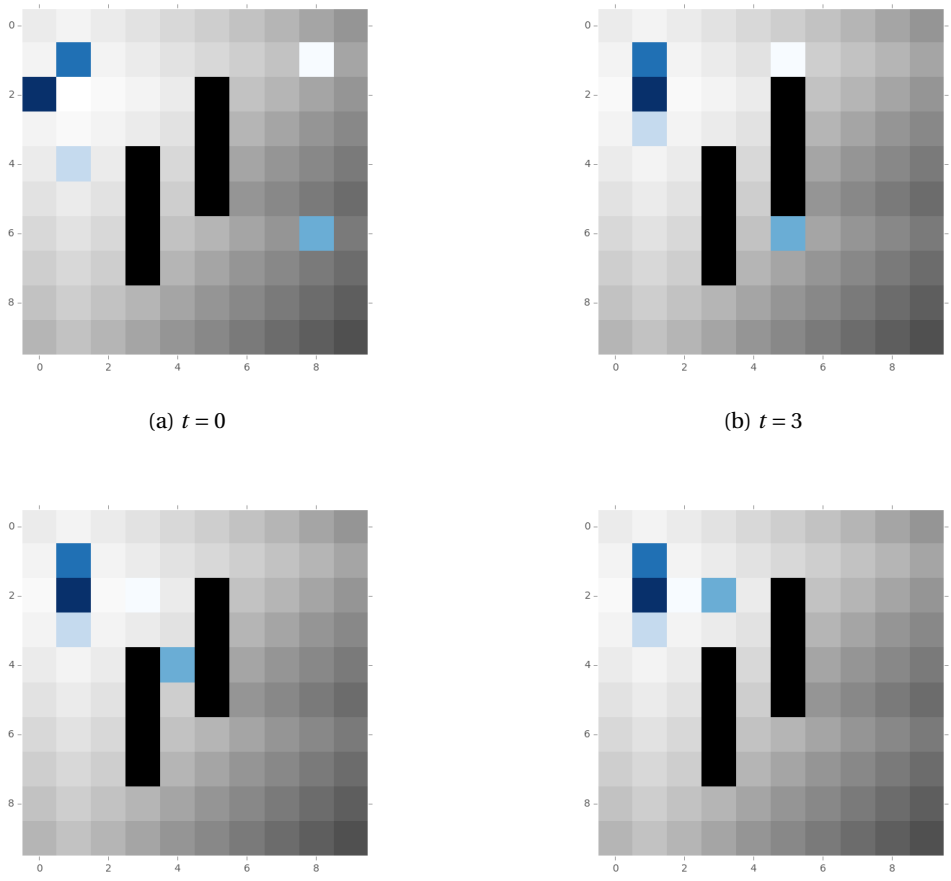


Figure 6.1: Example 1 from $t = 0$ to $t = 9$.

We can see our simulation performs as we would expect, with every person moving one spot per time step and finding the minimum path to the exit (which with no barriers and diagonal moves are not allowed will be "L" shaped). Notice the humans are piled by the exit, this is simply because we haven't removed humans from our simulation upon reaching the exit yet!

VERIFICATION TEST II: BOUNDARIES For our second test we implemented barriers. These can be walls, tress, or any other place on the map where a human is not allowed. Following the idea that the darkest spot on the grid is the furthest distance, we color walls true black. With the same human starting position as our first test, let's run a second test with two barriers.



(a) $t = 0$ (b) $t = 3$

(c) $t = 6$ (d) $t = 9$

Figure 6.2: Example 2 from $t = 0$ to $t = 9$.

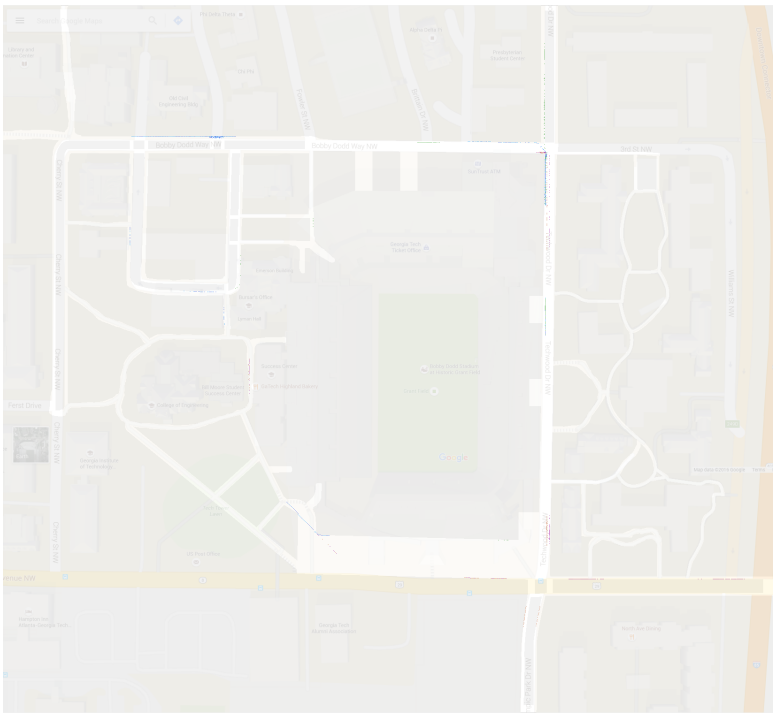
Just as we would expect, we see the top right, white-ish human moves left before down as to avoid the barrier. More clearly, we see the bottom right human move in between the walls and ultimately find the exit. Once again, these barrier can now be scaled to construct any shape

and represent areas where humans are not eligible to walk.

7 Final Simulation, Validation, and Qualitative Discussion

Visualization We developed a visualization code to read in console output from the model code and produce animations of humans walking over the domain. We will make extensive use of visualizations to argue the validity of our model. Since our simulation model defines multiple matrices representing the domain grid, the pedestrians, and other force fields, we want to be able to visualize each matrix overlayed upon one another. The visualizations currently include a discrete gradient representation of the static field, as well as blacked-out walls in order to watch humans avoid them. We color people by five colors, each corresponding to a specific exit.

Case Study I: 1,000 People We first present a simulation run with 1,000 people. We print out the number of people simulation for each time step to verify our constructed function spawns people according to our observed histogram. Once verified, we can view the output bitmaps transparently overlayed onto our map.



(a) 1,000 people: $t = 500$.

(b) 1,000 people: $t = 1000$.

When viewing the full scale of the map with only 1,000 people it is a bit difficult to see everyone, especially when each person is the size of one pixel. However, we were able to run this simulation a handful of time and calculate an average time that it takes the stadium to be evacuated. Assuming 1 pixel is 1ft^2 and the average person moves at 3 miles per hour, we calculate that each simulation-time step is approximately $\frac{1}{4.4} = 0.23$ seconds long. By varying the RNG seed we obtain completely different simulations since we assign exits uniformly randomly. We found an average exiting time of 1,000 people around 2000 time-steps, or approximately 7.6 minutes. This seems a bit quick, but with only 1,000 people it is not much of an evacuation as it is a normal day on campus!

CASE STUDY II: 25,000 PEOPLE We can now see the full simulation shine when we increase the number of people in the simulation. The output visualization better shows the types of images we are able to present.

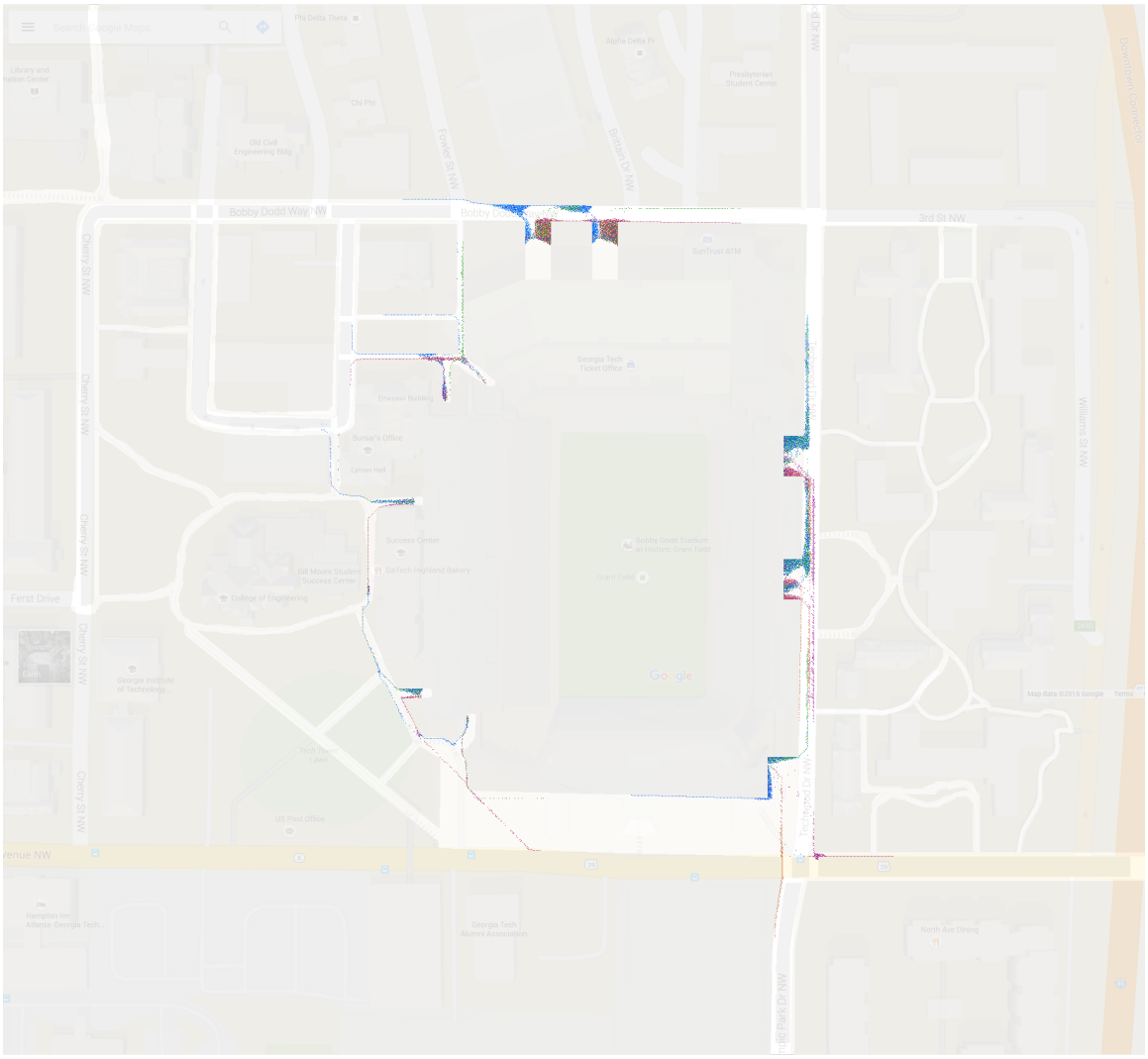
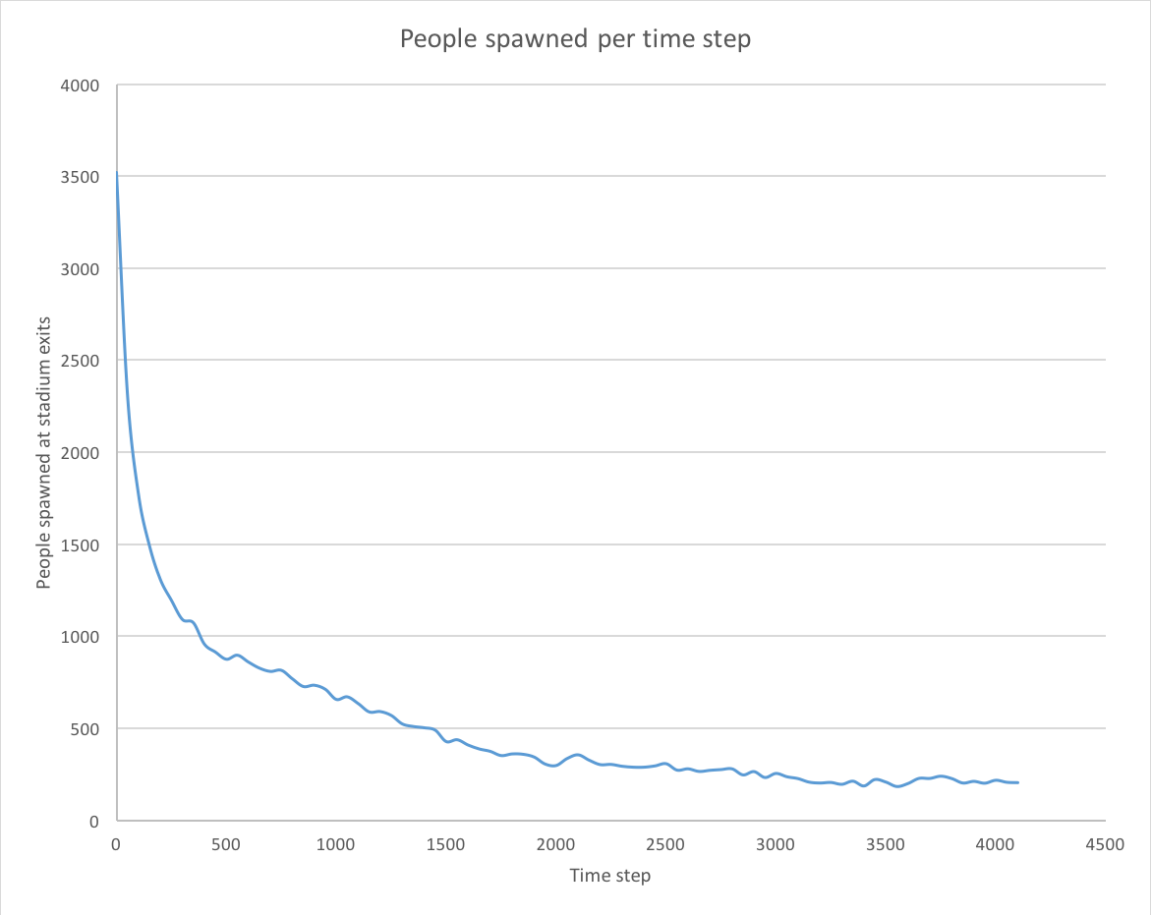


Figure 7.1: 25,000 people towards the beginning of a simulation.

Now we can see in the spawning locations where people are sorting themselves out, i.e blue people and red people clearly divide as they find their assigned exit. We also see individuals traversing diagonal paths in order to reach their destination quicker than only right angle turns. We also see people avoiding crowds by forming lanes in the middle on paths when the edges of walls are not accessible due to other people moving. We think these micro and local interactions adequately represent a physical system for the purpose of this class project, however we discuss improvements at the end of this report.

CASE STUDY III: 55,00 PEOPLE At 55,000 people we are simulating exactly how many people can fit inside of Bobby Dodd stadium. We can verify that indeed 55,000 people are being generated by our simulation by observing the following plot of the total number of people

spawned based on the current number of people spawned. Notice it follows closely to our exponential decay function.



When we look at the visualizations, we observe similar crowd dynamics in this case study as the previous, however we notice one problem in our simulation: gridlock.

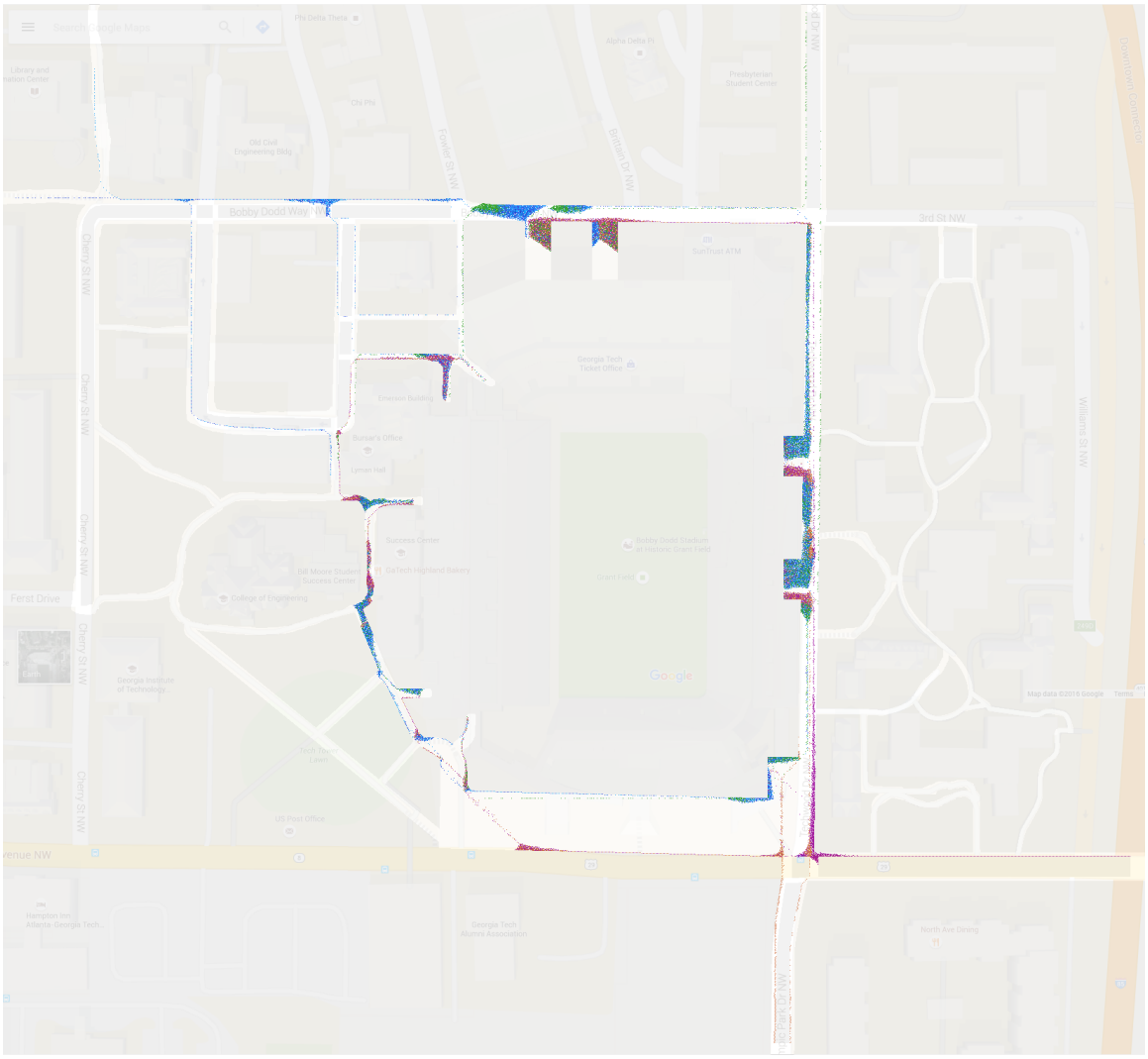


Figure 7.2: 55,000 people just trying to go home.

We expect intersections and narrow paths to be congested just as they are in reality, but due to the sheer number of people in this case study we see particular gridlocks form where and entire wall of red people cannot avoid an entire wall of blue people.

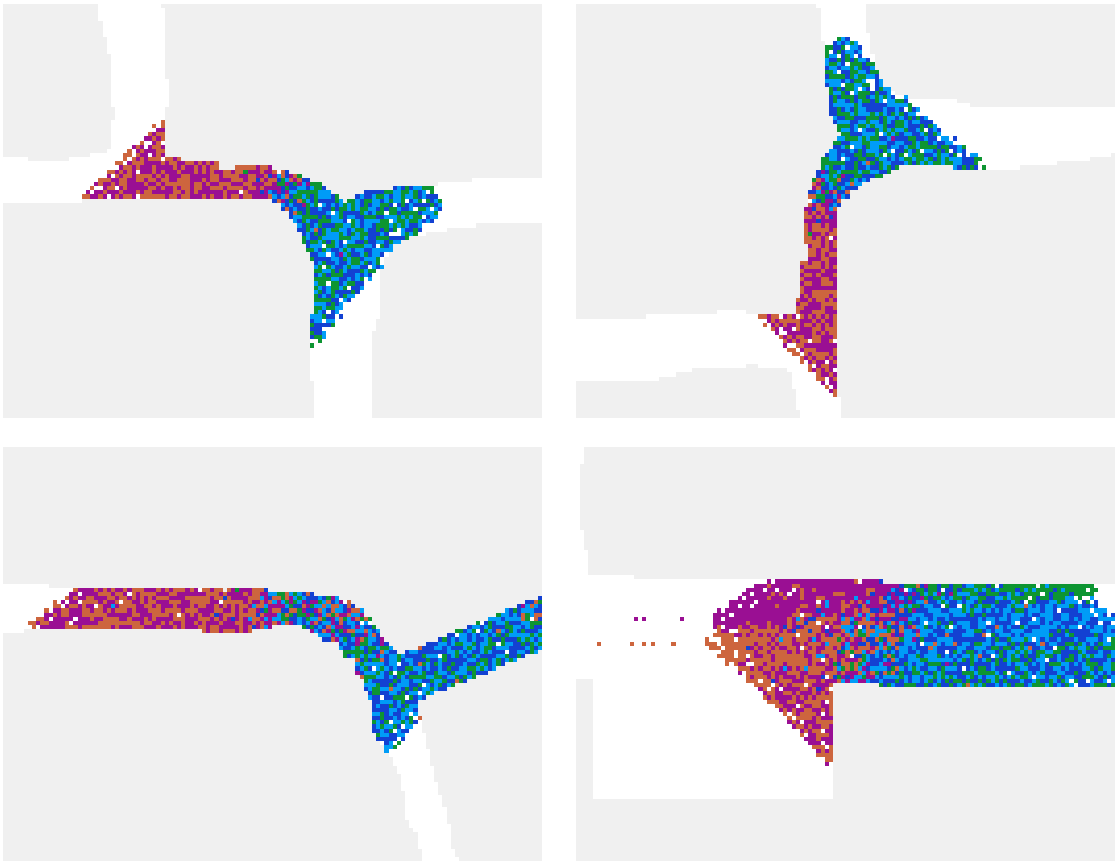


Figure 7.3: 55,000 people at grid lock.

8 Final Thoughts and Future Work

Currently we have an implementation of a simulation engine driving humans towards exits on a specified path network whose input parameters consists of the number of people to be simulation, a RNG seed for exit assignments and conflict resolution, and number of time steps to output visualizations. We presented two verification examples, as well as discussed the techniques used to perform visual validation. Furthermore, we are able to create fluid animations of our pedestrian evacuation that this report does not do justice representing in still images. Finally, we identify three ares that could benefit from continued work:

1. As stated before, we were not able to simulate the full 55,000 people as certain gridlocks formed on the smaller paths. We should note there was no computer hardware limitation, as we optimized memory consumption as well as we simulated 55,000 people on screen at once, but the paths were not wide enough or another rule needs to be added to address the complete gridlocks. A proposed solution could be to make the pinch points wider so more people can move around crowds.

2. We made extensive use of static and dynamic fields, and experimented with calculating them a few different ways (use only cardinal directions, make use of corners, etc.) however we were not able to implement full functioning anticipation field, rather we borrowed ideas to make more simplistic, heuristic rules. If given more time, priority would be given to developing this other force field to merge it with out existing fields and rules.

3. Due to time constraints, we were not able to perform as rigorous of an output analysis as we initially wanted. We believe our visualizations presented are representative of the hundreds of result images, however we would like to obtain more statistical evidence to support our conceptual model choices, such as an average evacuation time per exit per person.

Regardless, due to the modularity and structure of our code, extending the simulation to include more fields and more technical features is feasible and well-posed.

REFERENCES

[1] Burstedde, Carsten, et al. "Simulation of pedestrian dynamics using a two-dimensional cellular automaton." Physica A: Statistical Mechanics and its Applications 295.3 (2001): 507-525.

[2] Peacock, Richard D., Erica D. Kuligowski, and Jason D. Averill. Pedestrian and Evacuation Dynamics. New York: Springer, 2011. Print.

[3] Markowski, Michael J. Modeling behavior in vehicular and pedestrian traffic flow. ProQuest, 2008.

[4] Hindawi Publishing Corporation Mathematical Problems in Engineering Volume 2015, Article ID 308261, 6 pages http://dx.doi.org/10.1155/2015/308261Modeling Unidirectional Pedestrian Movement: "An Investigation of Diffusion Behavior in the Built Environment"

[5] J.theor.Biol(1993)160, 97-133 "Cellular Automta Approaches to Biological Modeling"

[6] "PEDESTRIAN TRANSPORTATION PROJECT PRIORITIZATION INCORPORATING APP-COLLECTED SIDEWALK DATA" A THESIS PRESENTED TO THE ACADEMIC FACULTY ALEXADRA FRACKLETON.

[7] Nowak, Stefan, and Andreas Schadschneider. "Quantitative analysis of pedestrian counterflow in a cellular automaton model." Physical Review E 85.6 (2012): 066128.

[8] Suma, Yushi, Daichi Yanagisawa, and Katsuhiro Nishinari. "Anticipation effect in pedestrian dynamics: Modeling and experiments." Physica A: Statistical Mechanics and its Applications 391.1 (2012): 248-263.

[9] Huang, Hai-Jun, and Ren-Yong Guo. "Static floor field and exit choice for pedestrian evacuation in rooms with internal obstacles and multiple exits." Physical Review E 78.2 (2008): 021131.

[10] Transportation at Georgia Tech. http://transportation.ce.gatech.edu/node/2602

[11] Goncalo's Home Page. Department of Biostatistics' Center for Statistical Genetics. University of Michigan. http://csg.sph.umich.edu/abecasis/

[12] Georgia Tech Athletic website. http://www.ramblinwreck.com