A Traffic Simulation Model Allowing For Wide-ranged Vehicle Communication

Timmy Galvin
Thomas Jefferson High School for Science and Technology
Alexandria, Virginia

June 10, 2009

Abstract

Traffic is an ever-growing problem as population around the world increases exponentially and with it, the number of drivers. Previously, fluid flow models have been used in an attempt to model traffic, but as has been recently discovered, only agent based models can accurately model a traffic scenario as small perturbations can have a butterfly effect and change the entire system.

Keywords: traffic, agent based modeling, computer simulation, efficiency, communication

1 Introduction

I want to create a traffic system that is extremely variable to allow for a great deal of variation and user-specification. The model needs to be tested against simple, proven results before becoming more complex. I need to have an accurate model of driver behavior and an ability for my program to collect data and analyze the traffic situation and congestion well. The system will include varying road systems and the vehicles on the road will have their own properties such as location, speed, acceleration, speed limit, and aggressiveness, some of which will be user-defined. While the simulation is made in an attempt to copy human behavior, like all other standard traffic models in use, it will be collision-free. The new focus of my project will be to determine the effects on certain variables such as initial velocity and traffic light timing on traffic congestions and travel time.

2 Background

Traffic jams are caused when drivers don't have access to information about road systems ahead of them. If the vehicles have information, or more so, if the actual road system itself has information, it can respond and react according to that information and change certain limitations on the vehicles to alleviate traffic. Currently there is research into optimization of traffic and variable speed limit areas for places with high frequency traffic jams. Using a model, the extent that variable speed limits and other changing traffic laws can be measured to see if implementing that technology is a viable solution to traffic jams. This would lead to a model being much cheaper to find out the outcome. My simulation that will model different traffic scenarios will depict traffic jams and use information freedom to respond according. This will be viewed using an overhead view of the system. I hope to show some results graphically on the change in flow versus density.

3 Development Sections

The World class that my environment is stored in holds an ArrayList of all the Vehicles so that it can access their information and location. Using these values it will be able to detect traffic jams and react to them by altering conditions. I have set up a basic simulation to do this, at the moment it is a two-lane system with traffic flowing in two directions moving at realistic velocity and acceleration and responding to a speed limit and vehicles around them. There is a variable in the program for traffic density that can easily be altered to change the number of cars. This program will be successful if it one, accurately depicts traffic flow by means of a traffic density versus flow graph, and two, if it can use information from the entire system to alter traffic laws.

In adapting my program, I changed it to allow for user input. First I did this on a basic level. This includes sliders I added at the bottom of the display so that the user can adjust values for rate of car flow, speed limit,

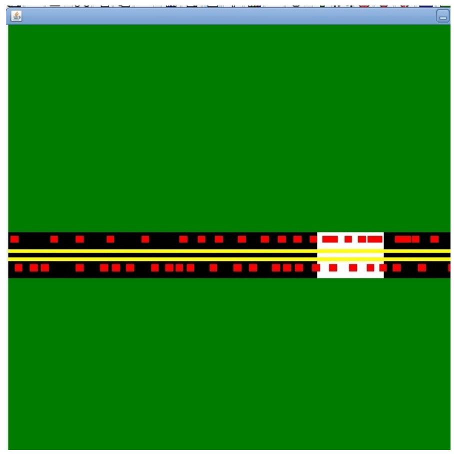

Figure 1: First-generation Road System with Speed Trap

speed trap limit, and acceleration rate. However, I found this wasn't enough user-defined information because I desired more adaptability.

Figure 1 shows what my first-generation

3.1 User-Generated Environments

In early versions of my program the environment is hardcoded into the system, to be drawn and to be reacted with by the cars. Because this severely limits the extents to which the program can be applied, I altered it to allow for users to create their own road systems which could be loaded into the program. The user would draw road systems in an imaging editing program such as Gimp and save it in pgm (ascii) format. This stores the data in a matrix-like fashion that can be easily read in and stored. The program reads this data in and scans the edges of the drawn environment to find roads. When it finds the start (or end) of a road, it traces it and stores its edge location to be used in car generation.

The two main black lines are lanes of a road and the grey separates a two-lane road. This system shows a single two-lane road. Different from the

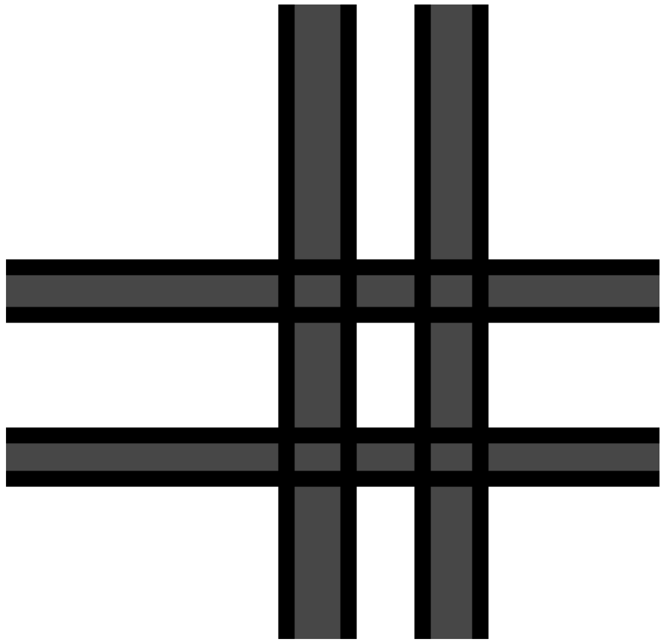

Figure 2: A New User-Defined Road Systems

far above picture, the images are now in gray scale to better suit an easier file format.

Since I have altered the focus of my program and abandoned curving roads and incorporate intersections a new potential road system would look like:

3.2 Scanning Algorithms

The scanning algorithm searches the edges of the environment to find the location of the roads. The code for scanning one edge looks like this:

```
public ArrayList scantXEdges()
{
ArrayList temp = new ArrayList();
int k=0;
int ig=0;
for(int x=50;x<650;x++)
{
```

```
if(worldmatrix[x][50]==0)
  if(ig==0)
  {
   temp.add(new Integer(x+11));
   ig=1;
   x+=16;
  }
  else
   ig=0;
}
return temp;
}
```

3.3 Locating Intersections

Identifying the intersections that the user has inputted was a major hurdle to overcome. The variability of this code is in that it can locate and handle intersections of any length and width. The code for this looks like this:

```
for(int a=50;a<650;a++)
{
if(worldmatrix[50][a]==0 && skip==0)
{
  for(int b=50;b<650;b++)
  {
    x=y=xx=yy=0;
    if(worldmatrix[b][a-1]==255
    && worldmatrix[b+1][a-1]==0)
    {
      x=b;
      y=a;
      for(int c=b+1;c<650;c++)
      {
      if(worldmatrix[c][a-1]==255)
      {
        xx=c-1;
        break;
```

```
      }
      }
      for(int d=a;d<650;d++)
      {
       if(worldmatrix[b][d]==255)
       {
         yy=d+1;
         break;
       }
      }
      intersections.add(
      new Intersection(x,y,xx,yy));
      skip=1;
      }
    }
  }
  else if(worldmatrix[50][a]==255)
    skip=0;
```

3.4 Reaction Algorithm

This is a main component of my program as it determines the braking speed of the vehicles in relation to the vehicle in front of them. At first I used a linear model but found it be rather inaccurate, or rather, that with my micro model, I could use more complicated functions because of small set sizes and the ability to visually observe the results. Currently it is a combination of two parabolic functions that vary the speed of the car behind as a function of its current speed and the speed of the car in front of it. This will prevent the car behind from ever running up on the car ahead of it or overrunning it.

This algorithm will need to be extended to include tailgating and aggressiveness. I will probably use previous research to base these algorithms off of since they will be more accurate than anything that I can come up with off of my visual perceptions of traffic.

The new updates with intersections now have the cars not only sensing for other vehicles but looking around them for intersections. And by accessing the states of the intersections, the vehicles can learn what color the light is and then act accordingly.

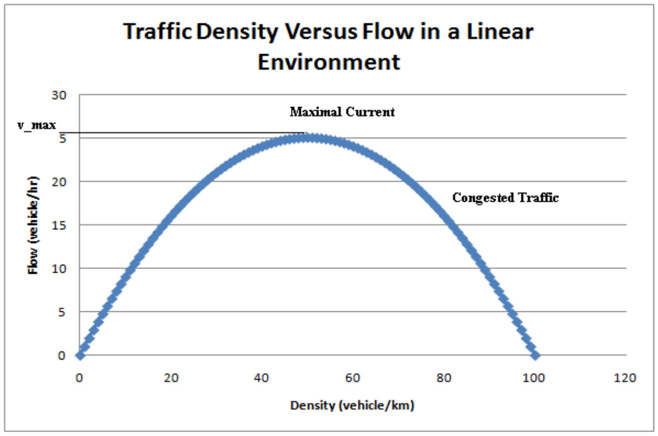

Figure 3: Density versus Flow

3.5 Graphing

Beyond the 2d system that I draw on display, I collect data to analyze about the traffic model. The data is then put into a graph of vehicle density versus vehicle flow. This graph will allow me to see if my model accurately describes traffic flow as is modernly accepted. My first generated graph for the linear model looks like this:

3.6 Intersections

A major addition to my project that adds a great deal of complexity is the addition of intersections. Allowing the user to define the road systems with any number of intersections increases the raw permutations. Also, the new focus of my project will be determing the influence of a number of variables on traffic congestion and travel time. The intersections are created with a standard stop light system. The stoplights also have the option of not being synchronized and the length of the red, yellow, and green lights are randomly generated within a certain range.

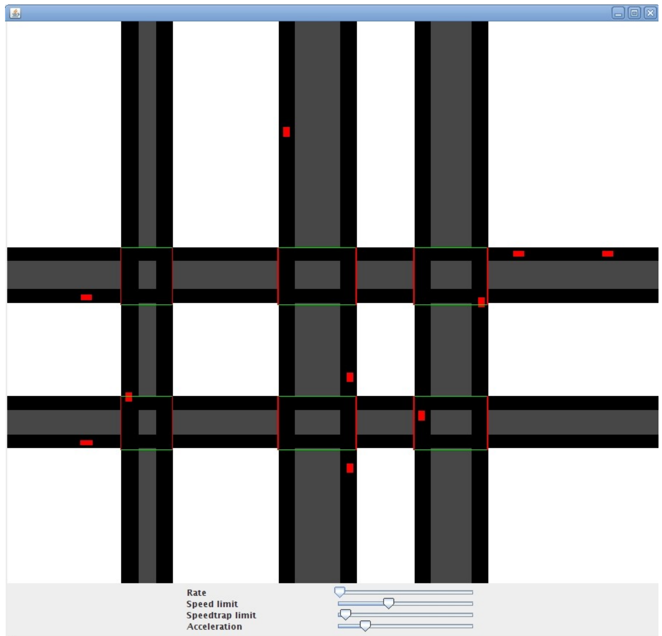

Figure 4: A New User-Defined Road System Being Implemented

4 Results and Discussion

The first generation of my program works in a two-lane traffic simulation. The only test of its validity that I have used is a visual comparison to known behavior. The behavior that it depicts is a traffic jam moving backwards in traffic. With the speed trap I built in, I was able to discover that it is better to have a speed trap earlier on rather than later, as traffic will build up on the road with the speed trap later on, making a slower lane. Hopefully, when my project is fully implemented I will yield results on how information sharing and system reaction will affect traffic flow, though I am not sure if it will be marginable enough to be reasoning behind buying many sensing systems for actual road systems.

Realistically allowing the user to define environments initially provides no results to my project because it still lacks the eventual elements that will separate it from other traffic simulations such as a 2d lane changing and detailed human characteristics. It is more steady progress that will eventually peak with a capable traffic simulation that allows for very extensive user definition.

I advise that my model be further developed and used as starting point. Current traffic simulation models are built upon micro models, a large compilation of them. If my code can be optimized and a computer network utilized, my program can be slightly altered to set up a large set of systems adjacent to one another to depict more than just a small road system, perhaps even a whole system.

References

[1] Bonabeau, Eric, “Agent-based modeling: Methods and techniques for simulating human systems”, Icosystems Corporation. Icosystem Corporation. 2002. 15 January 2008. ¡http://www.pnas.org/content/99/suppl.3/7280.full¿

[2] Dumbuya, et al., “An Agent-Based Traffic Simulation Framework To Model Intelligent Virtual Driver Behaviour”, *Driving Simulation Conference (2002)*, Paris, pp. 363-373.

[3] Ero, Kuhtlhow, “Application of Agent Technology to Traffic Simulation”, 2006. 20 January 2008.

[4] Khalesian, M. and Delavar, M. R, “A Multi-agent Based Traffic Network Micro-simulation Using Spatio-temporal GIS”, Center of Excellence in Geomatics Eng. and Disaster Management, 2008. 10 January 2008. ¡http://www.isprs.org/congresses/beijing2008/proceedings/$2_pdf/1_WG$- $II - 1/06.pdf >$.

[5] Nagel, Kai and Steen rasmussen, “Traffic at the Edge of Chaos”, Proceedings of the Alife IV, 2005. 31 October 2008.