Traffic Simulation

Ann Tupek
University of Alaska Fairbanks
CS 681

Abstract

This traffic simulation uses a single-lane car following model in which the drivers act identically except for their desired speeds to simulate traffic at a freeway cloverleaf interchange. Despite the small number of variables used, the cars in the simulation reflect real-life traffic situations such as congestion at the on-ramps and dispersal at off-ramps.

Introduction

There are many methods to simulate traffic. Macroscopic models describe traffic flow as analogous to the flow of liquids or gasses and look at the combined features of traffic such as density and average speed. In contrast, microscopic models describe individual vehicles that interact with each other and are similar to a particle system.

Within microscopic models are car-following models and cellular automata. In cellular automata, the space of the simulation is divided into discrete cells which are either occupied or unoccupied by a vehicle and the cell's state is determined after each time step. The car-following model utilizes differential equations in which the continuous variables such as speed and location depend on the time variable and the behavior of the front car.

These microscopic models can incorporate many choices for variables and driver behavior. Drivers can display identical or different driving behaviors, and their behavior can change based on the current situation in the traffic simulation. Single lane models and multiple lane models are also possible.

For this traffic simulation I have selected a car-following model. The drivers follow identical rules, however their desired speed is one of three available options. I have also selected a single-lane longitudinal model as opposed to a multi-lane lateral model. Instead of the drivers having set paths, when they reach a fork in the road their choice is randomly selected.

Materials & Methods

This simulation was written in Processing.js. I selected a map of two freeways that intersect with a cloverleaf interchange. I overlaid the paths onto this map and then broke up the paths into individual segments at whose endpoints traffic could either merge in or break away. I essentially ended up with a digraph.

Each car has a desired speed, randomly assigned at their creation, of fast, medium, or slow. Also randomly assigned at creation, each car has a beginning path segment and distance along that segment.

At each time step, each car checks the other cars along its segment to determine the distance between them. This distance is then compared to the car's speed times its reaction distance. If there is a car inside this threshold, the car slows down. Otherwise, the car attempts to speed up to its desired speed. For the purposes of this simulation, all cars have the same reaction distance, acceleration, and deceleration.

Results & Discussion

Though the initial placement of the vehicles along the segments is random, it is not long before groupings of cars form. These are led by the slow-moving vehicles. Congestion and bottlenecks tend to occur at the on-ramp merges and continue behind the slower-moving vehicles until turn-offs are available, at which time the congestion disperses. When a slower-moving vehicle turns off, the drivers that have a faster desired speed accelerate until they reach that speed or are blocked by another slower-moving driver.

The on-ramp merging isn't perfect, and when one line of cars merges in with another line of cars, the individual vehicles get shuffled together and appear to have collisions. This could be solved by adding additional lanes for the multi-lane segments of the highways and incorporating lateral lane-changing rules into the model.

Adding a 'look-behind' function could also alleviate the apparent collisions at the on-ramp merges. In addition to the follows information, each segment would also have feeds information, and would check the segments that feed into it for approaching cars and adjust the speed of the vehicle appropriately.

Changing the response times and look-ahead distances of the various cars to be different would further reflect real-life driving conditions. The number of different variables that can be added to a traffic simulation make it increasingly complex.

This simulation is not very complex, using only single-lanes and three different desired speeds. However, the model displays some real-world traffic attributes, with cars piling up behind slower moving vehicles and bottlenecks appearing at merges.

Sources

Kerner, Boris S. *The Physics of Traffic: Empirical Freeway Pattern Features, Engineering Applications, and Theory*. New York: Springer, 2004. Print.

Treiber, Martin. *Traffic Flow Dynamics: Data, Models and Simulation*. New York: Springer, 2013. Print.