A Simulation Design Based on HGE Towards a Manhattan Network Solution

Shuai DOU, Jin-tao XING*, Shuo-ran SHI and Xu-dong CHEN
School of Software Engineering, Beijing Jiaotong University, Beijing 100044, P.R. China
*Corresponding author

Keywords: Traffic simulation software, C++, System design, Visualization.

Abstract. The computer simulation of road traffic is an important tool for analysis and control of actual or designed road traffic networks. This paper introduces how to implement a road network simulation software by using C++, also contains architecture and the software system design process in detail. The software suits to Manhattan grid road network and we hope it can suit to other circumstances as well in the future. The result shows that this simulation software can support large amounts of data, visualization, more adjustable parameters.

Introduction

Traffic systems are an excellent application environment for simulation based research and planning techniques, an application area where the use of analytical tools.[1] The computer simulation of road traffic is an important tool for analysis and control of actual or designed road traffic networks.[2] Constructing microscopic traffic network simulation is also the most time-consuming portion during the simulation period. On the one hand, the physical network drawn needs to reflect the real geometric state of the road network; On the other hand, it is necessary to set realistic traffic rules on the simulation network. Currently commercial software of traffic simulation, such as VISSIM[3], PARAMICS[4], TSIS[5], has the ability of meeting most of the simulation needs, but researchers often cannot modify the model in the software due to its closed-source characteristic. As a result, scholars in the transportation field may cost more time in their research. For example, if they want to achieve the simulation according to their research model, they may not find a suitable commercial software. Besides, at present, many traffic simulation papers only focus on the results of simulation and the discussion about them. Therefore, it is sometimes extremely difficult for them to implement their own specific simulation software. This article will focus on how to design and implement a versatile software in an easy way, which provides a guideline to achieve simulation software they really need in a short time.

Due to the background above, this paper implements a road network data model base on Manhattan grid, and achieve the visualization of the simulation road network by C++, which support large amounts of data, visualization, more adjustable parameters. Most importantly, this paper describes the software system design process in detail.

This paper is organized as following. The architecture of our model present in Section2. The performance of this simulation model show in Section3. The conclusions are shown in Section 4 followed by discussion and future work.

The Architecture of Our Model

Each traffic simulation software architecture components are inseparable from the settings of car, road and lights. In this system, the overall structure of our road network is the Manhattan traffic network, and at the same time, the road network is more consistent with the actual traffic situation through classifying road sections. We use the NS rule to update the state of the vehicle. In the aspect of traffic lights, we set the traffic lights as the intersection’s inner class to achieve personalized configuration of each intersection traffic lights. Finally, our innovative use of HGE 2D game
rendering engine achieves efficient visualization of data dynamic display. In order to meet the experimental requirements of multi-road and large-scale vehicle data, we choose to use the highly efficient C++ as our programming language. Next, we will perform a detailed description of each part.

![Figure 1. Define a unique road.](image)

**Constructing the Road Network**

In the process of constructing the road network, how to realize the construction of road and the definition of different road’s relative location in the global road network are the difficulties when we design. The Manhattan Network is a choice for most traffic simulation software to build the basic road network. Through the Manhattan network, we can achieve the construction of road. We can use the location of two intersections to define each road’s location in the global road network. We give each intersection a two-dimensional coordinate to represent location of the intersection, and any road start and end at one intersection, so we can use the formula: (a, b) -> (c, d) to define a unique road (Figure 1).

In addition, according to their relative order starting from intersection to the next intersection, road can be divided into three parts: Entrance section, free section, exit section. A vehicle cannot change at the entrance and exit section. The division of different road segments is to simulate the actual process when a vehicle enters and leave the road section. Due to driver's driving habits and local traffic rules. When the vehicle enters and exits sections, drivers generally will not change the lane.

**Update the State of Vehicles**

We use NS rules[6] to achieve vehicles’ operation. The NS rules are often used in the traffic simulation model to update the vehicles’ state. The basic content of this rule is listed as follow.
The acceleration rule is:

\[ v_{t+1} = \min(v_{t+1}, v_{\text{max}}) \]  

(1)

The deceleration rule is:

\[ v_{t+1} = \min(v_t, d_t) \]  

(2)

The random stop rule is:

\[ v_{t+1} = \max(v_{t-1}, 0) \]  

(3)

The movement rule is:

\[ x_{t+1} = x_t + v_t. \]  

(4)

Where \( v_t \) is the maximum velocity of vehicles, \( x_t \) is the position of vehicle at \( t \) moment. In order to simulate sudden deceleration of the vehicle while driving, we introduce the third formula. The probability of this deceleration is \( p \) \( (p \neq 1) \). What we need to explain in particular is that this rule is not suitable for the leading vehicles. You should consider the corresponding traffic light and the crowded degree in the crossroad in order to update the state of the leading vehicles.

Figure 2. Rules for turn left, straight, turn right and turn-off lanes at the intersection.

In order to avoid vehicle conflicts at the intersection. We set new rules for turn left, straight, turn right and turn-off lanes at the intersection (Figure 2). Vehicles travels in a particular road of the corresponding direction, so there will be no conflict.

Setting the Traffic Light and Realizing Visualization

The impact of customized intersection traffic light control to the whole road network is one of the important topics in the field of transportation research, so it is necessary to achieve the intersection traffic light timing scheme which can be customized. We use object-oriented design ideas which commonly used in software development and set traffic lights as an inner class of an intersection. Different control strategies to achieve personalized custom timing program are through the personalized set of traffic lights with the inner class parameters.

The position of the vehicle in the road network is dynamically displayed according to the change of time, which enables researchers to experience the changing laws of vehicle distribution and vehicle
congestion intuitively. In our model, we use the game rendering engine HGE creatively for rendering roads and vehicle per second. HGE is a very popular open source hardware accelerated 2D Game Engine. It is a feature-rich middleware, which is based on DirectX8.0. Its architecture consists of three levels of abstraction: core functions, helper classes, authoring tools. Its packaging is so good that you only need to focus on the logic of the objects you want to draw and the changes that render between each frame, without the need to focus on internal implementation mechanisms such as DirectX and windows message loops. It is very easy for people who are not familiar with programming to use. Most importantly, it’s efficient use of memory make it possible to process the display of a large number of vehicles.

![Figure 3. Multi-road by zooming the road network.](image)

![Figure 4. Support large-scale vehicle data.](image)

**The Performance of This Simulation Model**

After experiencing the software system analysis, architecture design, coding implement, we tested and analyzed the performance of our simulation software. The experimental results show that our software can well meet the needs of our simulation requirement. What we need to particularly mention is that this model can support multi-road (Figure 3) and large-scale vehicle data (Figure 4). This is due to that we choose the Manhattan traffic network and the NS rules as our theoretical basis. What’s more, innovative use of efficient HGE game engine for dynamic rendering is the biggest bright spot in our model.
Conclusions

In this paper, we introduce how to achieve a traffic simulation software by constructing our model, setting the traffic light and finishing visualization. This software simulation suits to Manhattan grid road network. It also has easy-to-use, maintainable and extensible characteristics.

We use several circumstances to test it. The result show that it is a reliable tool.

In the future, we will use it to solve real problem in many other places. And we hope it can suit to more kinds of circumstances in the future. We will also improve system architecture and make sure the design easier to be customized.

Acknowledgement

This work is funded by the National Natural Science Foundation of China (grant nos. T17A200011) and the BJTU student's innovation project (grant nos.170185019, nos.170185010).

References


