

基于多智能体的交通控制仿真系统

熊继鹏¹, 张 俊¹, 娄联堂^{2,3,*}

(1. 武汉工程大学计算机科学与工程学院, 湖北 武汉 430074;

2. 武汉工程大学理学院, 湖北 武汉 430074; 3. 智能机器人湖北省重点实验室, 湖北 武汉 430074)

摘 要:研究了基于多智能体的交通控制 Agent 模型, 将多智能体技术(Multi-Agent)引入智能交通控制系统. 首先介绍了系统各功能模块的逻辑结构, 然后详细介绍了基于泊松分布、AABB 碰撞检测、粒子系统的 Agent 主体构建过程. 最后以 C++ 和 OpenGL 实现系统仿真, 系统能较好的模拟交通控制.

关键词:多智能体; 碰撞检测; 交通控制仿真

中图分类号:TP391.9

文献标识码:A

doi:10.3969/j.issn.1674-2869.2010.12.021

0 引 言

交通控制的智能化已成为人工智能领域最活跃的研究主题之一. 传统交通信号控制策略是定时控制, 是一种开环控制方法, 其适应性较差; 感应控制是一种闭环控制方法, 它可根据车辆信息检测绿时变化, 具有一定的适应性. 传统交通控制策略与方法固有的局限性, 迫使人们采用智能程度更高的技术来解决控制问题. 由于交通控制拓扑结构的分布式特性, 使其很适合于应用 Multi-Agent 技术, 尤其对于具有剧烈变化的交通情况, Multi-Agent 的分布式处理和协调技术更为适合^[1]. 本文从 Multi-Agent^[2]的角度研究智能交通控制系统, 对各 Agent 主体进行研究, 并最终完成系统仿真.

1 系统简介

通过对实际交通系统^[3]简化建立多 Agent 模型后完成的 Multi-Agent 仿真系统, 主要包括车辆随机生成 Agent、路口碰撞检测 Agent、车辆 Agent、系统调度 Agent 和交通流 Agent^[4]. 此外还有一些功能性的模块, 主要有外围环境渲染模块、中心动画生成模块、系统时间显示模块. 这些功能模块相对独立, 主要完成一些细节, 如纹理生成、时间显示等. 系统调度 Agent 在系统启动时根据用户选择启动车辆随机生成 Agent, 推进仿真钟, 协调各 Agent 的动作. 车辆随机生成 Agent 可

以根据以往统计数据或者随机生成模型(依据用户选择)生成车辆 Agent 数据. 交通流 Agent 用双链表记录相应车辆 Agent 数据, 反馈控制车辆随机生成 Agent 的参数(也可人工选择), 当车辆 Agent 驶离时, 释放其资源. 在仿真钟的触发下, 车辆 Agent 根据当前交通流 Agent 的数据作出相应动作, 调整速度、加速度、行驶方向, 是否停车等. 当车辆 Agent 到达路口时, 启动路口碰撞检测 Agent, 各车道车辆 Agent 进行自主协调(以多线程技术完成), 完成车辆 Agent 的路口通行. 系统能完成交通控制的基础仿真, 对多智能体交通控制系统的设计具有重要意义.

2 Agent 设计

2.1 车辆随机生成

借鉴排队论^[5]中到达时间间隔的分布, 建立了基于泊松分布的车辆随机生成 Agent 模型. 车辆产生模型描述了车辆的随机到达数分布, 车辆到达数的分布是离散型的随机分布. 车辆到达分布数常见的有泊松分布、平均分布及二项分布等. 其中泊松分布用于描述计数时间间隔极短, 车流密度不大, 车流间相互影响较小, 其他外界干扰基本不存在的交通状况; 平均分布用于描述车流比较平均的情况, 各车道产生车辆的概率相当, 二项分布用于描述车辆比较拥挤, 自由行驶机会不多的车流. 本仿真模型主要用到泊松分布模拟车辆产生的间隔, 泊松分布基本公式是:

收稿日期:2010-08-07

基金项目:国家自然科学基金项目(60975011)资助

作者简介:熊继鹏(1987-), 男, 湖北荆州人, 硕士研究生. 研究方向:图形图像分析与智能计算.

指导老师:娄联堂, 教授, 博士后, 硕士研究生系指导老师. 研究方向:图形图像分析. * 通信联系人

$$p(x)=\frac{m^x}{x!}e^{-m} \tag{1}$$

式(1)中: $p(x)$ 为在计数间隔 t 内 到达 x 辆车的概率; x 为给定时间间隔到达的车辆数; e 为自然对数的底数, $e=2.718\ 28$; $m=t$ 为计数间隔 t 时段内平均到达的车辆数, $\lambda=\frac{N}{3\ 600}$, N 是小时交通量, t 为每个计数间隔持续的时间(s).

2.2 路口碰撞检测

根据碰撞检测理论^[6],将车辆行驶的环境按照坐标值划分为网格来进行碰撞检测,即空间剖分法.在真正进行碰撞检测时,对车辆矩形运用 AABB(Axis Aligned Bounding Box)包围盒.碰撞检测的一般框架如图 1 所示.

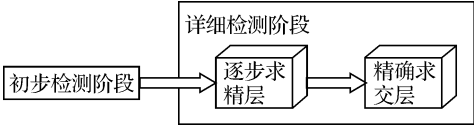


图 1 碰撞检测的一般框架

Fig.1 General framework for collision detection

按照碰撞检测的一般框架,经过初步检测阶段分析,4 个车道(2、5、8、11)的车辆不会发生碰撞,其余 8 个车道的车辆可能会发生碰撞.而在详细检测阶段的逐步求精层中,这 8 个车道的车辆碰撞检测又分为 2 类.在精确求交层将对这 2 类进行具体求解,分析如下:

1) 处于中间车道即 1、4、7、10 的车辆碰撞检测,以 1 车道为例进行分析.1 车道行驶的车辆是笔直通行的,可能在图 2 中 4 个三角红点处发生碰撞.图 2 中绿色菱形点为可能会发生碰撞点.车辆不一定刚好位于这些点,将这些点的前后各一格也标识为碰撞区域,1 车道有 25 个碰撞点.中间车道的车辆在检测点互不依赖,不会发生死锁.因而这 4 个车道的碰撞检测是独立进行的.

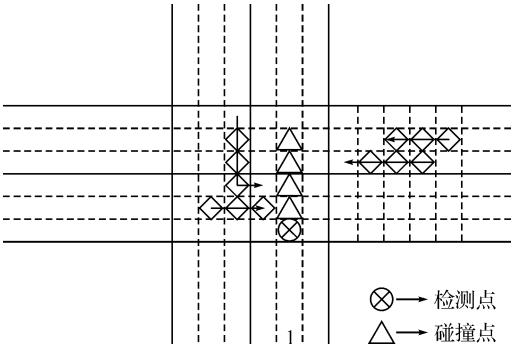


图 2 1 车道车辆碰撞检测分析

Fig.2 Analysis of vehicle collision detection in channel 1

2) 处于内侧车道即 0、3、6、9 车道的碰撞检测,以 0 车道为例进行分析.内道的情况要比中间

道的碰撞检测情况复杂的多,图 3 中虚线的菱形原本也是碰撞点,因行驶方向一致,变为非碰撞点.图 3 中的蓝色菱形是其他内道的检测点,这样可以看出内道之间会发生死锁,不仅如此,内道与中间道之间也会发生死锁,由进程优先权的思想,给 4 条内道赋予一个优先权,0、3、6、9 依次降低,而且内道优先权高于中道.经过这样的一个策略设计,内道的车辆将不会发生碰撞.

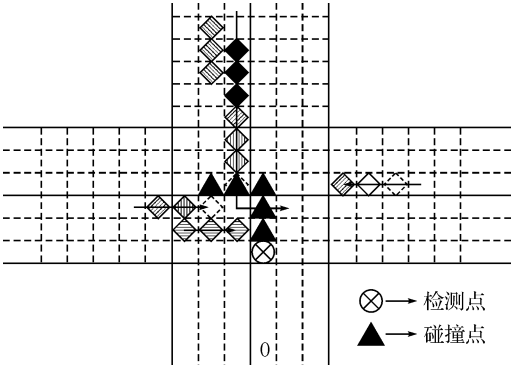


图 3 0 车道车辆碰撞检测分析

Fig.3 Analysis of vehicle collision detection in channel 0

2.3 系统调度

系统调度 Agent 根据仿真钟协调各 Agent 的动作,也即采用分时控制模型.每隔 20 ms 时钟步从全局进行一次车辆随机生成、碰撞检测、车辆 Agent 行为、交通流 Agent 更新操作.采用 C++ 及 OpenGL 实现 AnimationVehicle 函数如下:

```
void AnimationVehiele(int value)
{
    CollisionDetect();
    Node<VehicleRect> * q;
    q=Vehicle1. sit. first->next;
    while(q)
    {
        q->data. CalDistance();
        q->data. ChangeDirect();
        q=q->next;
    }
    DeleteLevaveVehicle();
    glutPostRedisplay();
    glutTimerFunc(20, AnimationVehicle, 2);
}
```

2.4 中心动画与车辆 Agent

中心动画与车辆 Agent,其实就是粒子系统.粒子系统通用结构如图 4 所示.

基于粒子系统理论^[7],中心动画粒子系统的粒子源位于坐标原点(0,0,0)处,角度增量是 15°,粒子数 24,颜色可不断变化.粒子生命期为 20 ms.

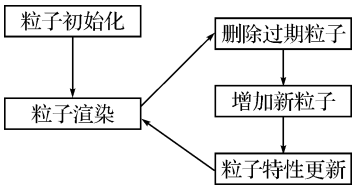


图 4 粒子系统结构图

Fig. 4 Structure of particle system

粒子属性更新由如下圆参数方程控制：

$$\begin{cases} x = r * \cos\theta \\ y = r * \sin\theta \end{cases} \quad (2)$$

其中 r 为粒子产生半径,根据一定步长变化, θ 为粒子角度,范围为 $[0,360]$,变化步长为 15 度。

车辆 Agent 由于数量较多,也构成了粒子系统. 车辆 Agent 粒子系统的粒子源共有 12 个,即 12 个车道的起点处,根据车辆随机生成 Agent 的数据生成车辆 Agent,所以每时每刻的车辆总数(即粒子总数)是不确定的. 粒子的属性每隔 20 ms 更新一次,其坐标点控制方程如下：

$$s = s + v_0 + \frac{1}{2}at \quad (3)$$

式中, v_0 为车辆初始速度,其正负由行驶方向决定,即规定向左和向下为负,向右和向上为正,如图 5 所示. 加速度根据车辆类型决定,若加速度为 0,则为匀速运动。

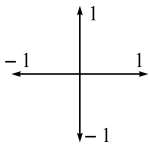


图 5 车辆行驶方向示意图

Fig. 5 Diagram for direction of vehicle

2.5 交通流 Agent

交通流 Agent 则从全局角度记录当前系统中所有车辆 Agent 的数据,方便其他 Agent 交互时的通信. 交通流 Agent 用双链表记录相应车辆 Agent 数据,反馈控制车辆随机生成 Agent 的参数(也可人工选择). 当车辆驶离系统时,释放其占用的系统资源。

3 仿真实现

系统模拟的是单个交叉口的交通控制. 车辆 Agent 的随机生成依据统计数据或者随机生成模型. 生成的车辆 Agent 数据记录在交通流 Agent 中,交通流 Agent 数据反馈控制车辆的随机生成,从而减少交通拥塞. 各车辆 Agent 可以与全局交通流 Agent 交互进行加减速,超车,停车等相应动作. 在更大范围的多交叉口的交通控制中,单个交叉口的交通流数据可以作为全局范围的短时交通

量预测依据,反馈车辆生成.

在 VC 环境下,采用 OpenGL^[8] 图形库及多线程技术实现的交通控制仿真系统如图 6 所示. 系统能较好的模拟交通控制,对多智能体交通控制系统的设计具有重要意义。



图 6 交通控制仿真系统

Fig. 6 Simulation system for traffic control

4 结 语

基于 VC 和 OpenGL 图形库开发的仿真系统,验证了智能化交通控制系统研究的可行性,体现了多 Agent 技术的优越性. 研究的主要创新点如下：

1) 基于多 Agent 技术,综合应用了排队论、碰撞检测和粒子系统建立了交通控制多 Agent 模型。

2) 基于 VC 环境和 OpenGL 图形库,实现了交通控制仿真系统。

与其他交通控制系统不同的是,运用 Multi-Agent 方法将从全局的角度控制交通. 车辆之间交互性和协调性增加,更加智能化. 进一步的研究工作可以如下展开：

(1) 从全局从发,协调多个交叉口的交通控制,优化交通时延、拥塞,车辆选择交通路线更加智能。

(2) 比较其他网络控制系统,研究优化交通控制模型。

参考文献：

[1] 王艳华. 基于多智能体 MAS 的智能交通控制系统的研究[D]. 长沙:长沙理工大学,2007,4.
[2] 蔡自兴. 人工智能基础[M]. 北京:高等教育出版社,2005.
[3] 何晓鸣,李玉娟. 咸宁市公路交通需求分析[J]. 武汉工程大学学报,2010,32(5):80-84.
[4] 刘黎志,刘军. 一个基于消息通知的工作流管理系统[J]. 武汉工程大学学报,2008,30(2):94-97.
[5] 于志青. 排队论在交通工程中的应用研究[J]. 中州大学学报,2005,22(1):118-119.

[6] 李士明,曹凯. 交通仿真系统中碰撞检测技术研究 [J]. 农业装备与车辆工程,2009(2):30-33.

[7] 吴继承,江南. 虚拟现实 中粒子系统的设计与应用 [J]. 计算机仿真,2004,21(11):137-140.

[8] Richard S. Wright, Jr Benjamin Lipchak, Nicholas Haemel. OpenGL SuperBible, 4th Edition [M]. Boston: Addison-Welsley Publishing Co,2007.

Simulation system for traffic control based on multi-agent

XIONG Ji-peng¹, ZHANG Jun¹, LOU Lian-tang^{2,3}

(1. School of Computer Science and Engineering, Wuhan Institute of Technology, Wuhan 430074, China;
2. School of Science, Wuhan Institute of Technology, Wuhan 430074, China;
3. Hubei Province Key Laboratory of Intelligent Robot, Wuhan Institute of Technology, Wuhan 430074, China)

Abstract: A model based on multi-agent of traffic control is proposed. Multi-agent technology is brought into the intelligent system for traffic control. First, logical structure of each functional module is introduced. Then, poisson distribution, AABB collision detection and particle system are applied to build the agent object. At last, the system is well simulated with C++ and OpenGL.

Key words: multi-agent; collision detection; traffic control simulation

本文编辑:陈小平

☆
(上接第 34 页)

Preparation and characterization of TiO₂ N-doped by nitrogen plasma treatment

LI Chao-yu, LI Jun

(School of Chemical Engineering and Pharmacy, Wuhan Institute of Technology, Wuhan 430074, China)

Abstract: The self-made nanosized anatase TiO₂ powder was synthesized by hydrolysis of titanium ester (Ti(OC₄H₉)₄) in basic media and then treated by N₂ plasma. The effect of plasma treatment conditions on the activity of photo-catalytic degradation of the Cr₂O₇²⁻ was investigated. The samples were characterized by X-ray diffraction (XRD), X-ray photoelectron spectroscopy (XPS), and Ultraviolet spectra (UV), Infrared spectra (IR). The 399.6 eV peak on the spectra of XPS showed that there was a certain of N1s after plasma treatment, indicated that nitrogen on the TiO₂ surface can be formed through N—O replacement during N₂ plasma, the spectra of UV showed that the light absorption move to the visible region after plasma treatment. The spectra of XRD indicated the crystallographic forms and particles dimension were no apparently change before and after the plasma treatment. The photo-catalytic activity of plasma treatment TiO₂ was better than the un-treatment.

Key words: titanium dioxide; photo-catalytic; nitrogen plasma

本文编辑:张瑞