Journal of Southeast University (English Edition)　Vol.19　No.2　June　2003　ISSN 1003—7985

Visual object-oriented application for lane following on intelligent highway system

Wang Chunyan[1,2]　Wang Wei[1]　Lu Jian[1]　Chang Yulin[3]

([1] College of Transportation, Southeast University, Nanjing 210096, China)
([2] National Center of ITS Engineering and Technology, Beijing 100088, China)
([3] Automobile and Transportation Engineering College, Jiangsu University, Zhenjiang 212013, China)

Abstract: A visual object-oriented software for lane following on intelligent highway system (IHS) is proposed. According to object-oriented theory, 3 typical user services of self-check, transfer of human driving and automatic running and abnormal information input from the sensors are chosen out. In addition, the functions of real-time display, information exchanging interface, determination and operation interweaving in the 3 user services are separated into 5 object-oriented classes. Moreover, the 5 classes are organized in the visual development environment. At last, experimental result proves the validity and reliability of the control application.

Key words: intelligent transportation system; intelligent highway system; lane following; visual object-oriented application

Intelligent highway system (IHS) is a new solution to the existing traffic problems. It is the final target of intelligent transportation system (ITS), taking people, cars and road as a whole body. Some research results demonstrate that ITS, based on existing traffic infrastructure, can increase traffic flow by two or three times, reduce waste gas by 20% or 30% at least, and as a result improve traffic running efficiency and our living environment[1-4]. Lane following is one of the key technologies in the IHS research field. While much essential progress of IHS research has been made in America and Japan[1,3,4], some ground breaking progress has also been made in China[5-8].

Object-oriented programming (OOP) offers promising facilities for designing the application of lane following on IHS (LFOIHS). OOP can promote the working efficiency of programmers, make software maintenance easier, and improve the production capability of software development[9,10]. Visual C++ 6.0 is one of the main products of OOP languages. This contribution discussed how to make the application structure for LFOIHS in light of OOP rules, established the software in Visual C++ 6.0 developing studio, and finished the lane following experiment. At last this contribution verified the validity and reliability of the visual LFOIHS application by experimental results.

Received 2003-01-02.
Foundation item: The National 10th Five-Year Plan (2002BA404A26).
Biographies: Wang Chunyan (1966—), female, senior engineer, graduate, wcy@itsc.com.cn; Wang Wei (corresponding author), male, doctor, professor, weiwang@transtar.net.cn.

1 Control System for LFOIHS

The control system for lane following consists of five parts: road surface, magnetic marker, relative position sensor, the sample car and the computer on board (See Fig.1). The markers are embedded under the road surface. When the sample car runs on the testing course, the sensors collect and pass the relative position information to the computer, and then the computer determines what kind of turning operation it needs and sends corresponding commands to the servo device. By repeating the process, the given task is finished[2,8].

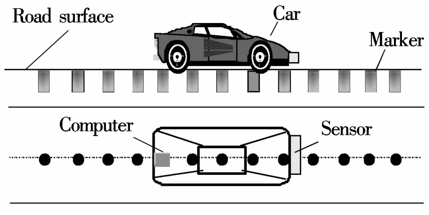

Fig.1 Control system for LFOIHS

2 OOP and Visual C++ Applied in LFOIHS Application

2.1 OOP technology applied in LFOIHS application

The core of the conventional function-oriented programming (FOP) is function-division. It organizes an application by data structure and function modules. Its fatal disadvantage is that once the data structure is to be changed, all the involved parts must make corresponding changes, which results in the FOP application's lower reusability and higher maintenance costs[9]. Completely different from FOP, OOP uses an application for a set of dispersed objects, involving the three concepts: object, class and message. In addition, an OOP application structure falls into two parts: class definition and class application. Furthermore the objects can do all the operations by way of sending or receiving various messages. In an OOP application, an object fixes its data and operations into a separated body. The objects can react to each other by message. The emphasis of OOP is the data.

2.2 Visual C++ technology for LFOIHS application

In a visual developing studio, every engineering project contains an application object of CuserApp. It is derived from the MFC class of CWinApp. If the application is made in the AppWizard (exe) type of multi-document, there are two more objects of CMainFrame and CchildFrame in the project. They are derived from the MFC class of CFrameWnd[9]. In a VC project, passing messages does both the linkage among the objects and the linkage between the system and the application, the class controlling data-save and data-display, and the view object of CUserView taking charge of display data. If the object of a Mainframe builds a standard frame on the screen, the object of CUserDoc creates a document and the object of CUserView creates a view[10]. It can be said that these main objects are involved throughout the course of calling various functions anywhere in the whole application.

3 Visual C++ Applications for LFOIHS

Some typical OOP methods used to make an OOP application are OMT, OOD/BOOCH, RDD/WRIFS-BROCK, OOAD/COAD-YOURDON, OOSE/JACOSON and VMT. Among the above methods, VMT method incorporates the advantages of the other methods, and introduces the visual method into the programming. VMT mainly considers questions of field and system architecture to support the solution at the design stage, and then involves design standards and analysis models to find the solution[10]. Thus, the LFOIHS application here was designed using VMT ideas.

In light of VMT, the participator is referred to any body in charge of exchanges between the system and outside. Here the participators of the LFOIHS application were the driver and sensors, and the source of the event response. User services were a series of exchanges between the system and participators. Here they were the three commands: self-check, manual drive and automatic drive, along with abnormal information collection from the sensors. Because the operations of responses to the abnormal information, display on the interface, and sending turn commands to the direction servo device, were all involved in the three user services, it was sufficient to take them into consideration.

The three user services of self-check, manual drive and automatic drive, and the abnormal information collection from the sensors are interconnected. For example, in the course of self-check, the on-board computer collected the state information of hardware parts, determined whether they worked normally or not, and then displayed the result on the interface. In addition, in the course of manual drive, the on-board computer collected the information about car-road relative position, determined whether it was suitable for transfer from manual drive to automatic drive, and then displayed the suitable transfer time on the interface. Furthermore, during the automatic drive state, the computer collected the data from sensors to determine what kind of turn commands needed to be sent to the servo device, and then displayed the run information on the interface. We know from the detailed analysis above that all three contained the two steps of data-save and display. Thus the two steps could be eliminated out as two common classes for their common characteristics, and then be used to finish the user services by way of exemplifying classes and defining message mechanics. As the user services must be demonstrated on more than one interface, two more classes of CMainFrame and CChildFrame should be added to the engineering project.

In the VC developing studio, this engineering project contained five main classes as follows: CUser-App, CUserDoc, Mainframe, CChildFrame and CUserView, among which CUserApp took charge of exchanges between the application and Windows, CUserDoc defined the data format for data saving and Mainframe, CChildFrame and CUserView were in charge of user interfaces. At the same time the class of CUserView also displayed data on the interfaces. Finally the project finished its job by way of creating some dispersed objects from these classes, defining the message mechanics and linking them together by messages. Because in the course of OOP, it was not the object but the class that was to be programmed, so far the application of LFOIHS was completed. Fig.2 shows the relationship among the objects in the visual OOP

LFOIHS application.

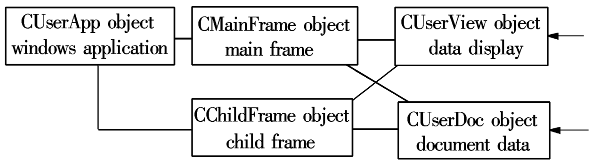

Fig.2 Relationship between the objects in the LFOIHS application

4 Experiment Proving

Equipped with the LFOIHS application, the sample car of BJ 2021 A6L jeep, made by Beijing Jeep Corporation Ltd., had been running on the IHS testing course in TCPG (Tong County Proving Ground) in Beijing, for more than 60 d. At last the application finished its given task, and its running speed was over 80 km/h at the direct section and over 40 km/h at the curve section. Moreover its success ratio overran 98%, and its experimental error was within the range of [− 20 cm, + 30 cm] with a reading error of less than ± 1.0 cm[8].

5 Conclusion

The experimental results show that the application can work efficiently and reliably. In order to promote the precision of the experiment, it is suggested to take higher precision position offset sensors and to make the controller more reasonable.

References

[1] Wang Jengyu. *Robust lateral control of heavy vehicles on automated highways*[D]. Berkeley: University of California, 2000.

[2] Wang Chunyan, Wang Xiaojing. Fuzzy controller for lane-following based on basic road establishment [J]. *Journal of Traffic and Transportation Engineering*, 2002, **2**(4): 90 − 94. (in Chinese)

[3] Yang X, Guo M, He K, et al. The study on the model of the autonomous mobile robot based on neural networks[A]. In: *Proceedings of 5th International Conference on Intelligent*[C]. Japan: Sapporo, 1998. 132 − 139.

[4] Becker J C, Simon A. Sensor and navigation data fusion for an autonomous vehicle[A]. In: *Proceedings of the IEEE Intelligent Vehicles Symposium* 2000[C]. 2000. 156 − 161.

[5] Wu Chaozhong, Wang Chunyan, Yan Xinping, et al. Information fusion of lateral displacement measurement in automated steering control system [A]. In: *Proceedings of the 9th ITS World Congress* [C]. Chicago: Mira Digital Publishing, Inc. 2002.

[6] Wang Chunyan, Wu Chaozhong, Ding Zhensong, et al. Study on hardware controlling system for lane-following on intelligent highway with embedded magnetic nails [J]. *Journal of Highway and Transportation Research and Development*, 2003, **20** (1): 129 − 132. (in Chinese)

[7] Wang Chunyan, Liu Zuo, Wu Chaozhong, et al. Design and implementation of control system for lane-following[A]. In: *Proceedings of the 9th ITS World Congress*[C]. Chicago: Mira Digital Publishing Inc,2002.

[8] Wang Chunyan, Li Bin, Liu Qingbin. Experiment study on lane following based on magnetic markers tracking [A]. In: *Proceeding of the 2nd Beijing International Exhibition and Seminar on ITS*[C]. Beijing: Science Press, 2002.

[9] Feng Yulin, Huang Tao, Ni Bin. *Object technology guidance* [M]. Beijing: Science Press, 1998. (in Chinese)

[10] Chen Weixing, Lin Xiaocha. *C++ object-oriented programming guidance* [M]. Beijing: Tsinghua University Press, 2000. (in Chinese)

面向对象的智能公路车道追踪可视化控制软件研究

王春燕[1,2]　王　炜[1]　陆　建[1]　常玉林[3]

([1] 东南大学交通学院, 南京 210096)
([2] 国家 ITS 工程技术研究中心, 北京 100088)
([3] 江苏大学汽车与交通工程学院, 镇江 212013)

摘　要　本文基于可视化面向对象理论设计了智能公路车道追踪计算机软件控制系统.根据面向对象理论,从各种相互联系的用户服务中抽象出 3 个典型的用户服务:人工/自动驾驶转换、系统自检、传感器异常数据收集,接着把贯穿于其中的数据实时显示功能、人机信息交互功能及判定与操纵功能分解为符合面向对象技术的 5 个类,并在可视化编译环境中把其组织成一个工程.最后实验结果论证了该软件控制系统的有效性.

关键词　智能交通技术; 智能公路系统; 车道追踪; 可视化面向对象应用程序

中图分类号　U491