A Prototype System for Real Time Computer Animation of Slow Traffic in a Driving Simulator

Jos B.T.M. Roerdink, Mattijs J.B. van Delden, Andrea J.S. Hin
Department of Mathematics and Computing Science, University of Groningen,
P.O. Box 800, 9700 AV Groningen, The Netherlands
Tel. +31-50-3633931; Fax +31-50-3633800;
Email `roe@cs.rug.nl`; URL: `http://www.cs.rug.nl/~roe`
and
Peter C. van Wolffelaar
Traffic Research Center, University of Groningen,
P.O. Box 69, 9750 AB Haren, The Netherlands
Tel. +31-50-3636764; Fax +31-50-363784; Email `wolff@trc.rug.nl`

Abstract

The Traffic Research Centre (TRC) of the University of Groningen in the Netherlands has developed a driving simulator with 'intelligent' computer-controlled traffic, consisting at the moment only of saloon cars. The range of possible applications would be greatly enhanced if other traffic participants could be simulated too. This paper presents a study of the possibilities of simulating bicyclists and pedestrians in general, and in the TRC driving simulator in particular. The new traffic participants must be controlled interactively, and move and behave in a natural way. A prototype system has been designed and implemented. The system simulates a bicyclist that is controlled by the user through a graphical interface. Integration of the prototype system in the TRC driving simulator is the subject of future research.

Keywords: Computer animation, driving simulator, slow traffic, motion generation

1 Introduction

The Traffic Research Centre of the University of Groningen has developed a computer controlled driving simulator, which provides a safe and programmable environment for road user studies [vWvW95]. The simulator is made up of a car installed on a fixed platform. Traffic scenery is generated by a Silicon Graphics SKY-WRITER 340 VGXT graphical computer with two hardware graphic pipelines, and is projected on the curved screen by three large video projectors, cf. Fig. 1. The traffic environment currently consists of 'intelligent' cars that interact with each other and with the simulator user. An important shortcoming lies in the fact that only

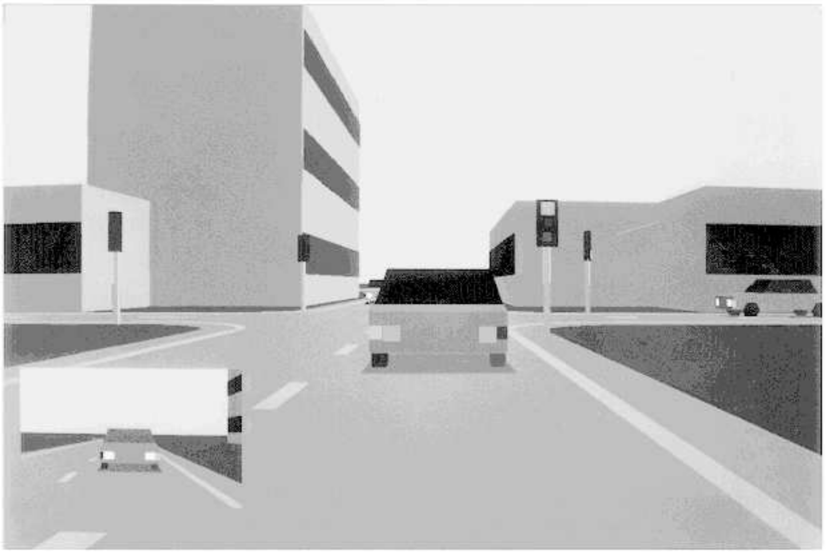

Figure 1: *Screen image of the middle projector of the TRC traffic simulator.*

saloon cars are simulated. An extension is desirable to pedestrians and bicyclists. These two classes of traffic participants are substantially different from the currently implemented cars: they have several parts that move with respect to each other, such as moving arms and legs. In this paper, we report on a prototype system for real time graphical animation of bicyclists and/or pedestrians, which will be collectively referred to as ViPs, or 'Virtual interactive Persons'. For more details the reader is referred to [RvDHvW96, vD95].

The prototype system has a number of requirements concerning speed, interactivity and naturalness of motion. First, it must be possible to simulate about 10 to 20 pedestrians and bicyclists in real time. This puts a severe constraint on possible solutions with respect to execution time. In addition, the visual simulation has to be interactive, meaning that any actions of the simulator user should be immediately reflected in changes of the state of the environment. Generally, delays up to 100 ms are considered acceptable in driving and flight simulators. The TRC driving simulator uses two types of frames: drawing frames and simulation frames that run asynchronous and in parallel. The graphical drawing process generates frames as fast as possible in an endless loop taking actual world position data of the simulated vehicles for each new frame from the simulation process. By extrapolating these positions in time vehicle motions appear very smooth on the screens. For pedestrians and bicyclists, however, extrapolating world positions in time will not be feasible as they are linked to gestural movements. This implies that all calculations necessary for displaying the next frame must be completed before drawing of that frame starts.

When dealing with pedestrians and bicyclists, humans must be modeled while walking, running and cycling. Accurate human models take up thousands of polygons and are not feasible for interactive simulation. However, realistic or natural *motion* will be crucial for making the models appear as humans. Also, every ViP needs to have its personal style of walking and cycling, called 'personification of the motion'.

The 'intelligent' vehicles in the TRC driving simulator are governed by a model which controls their behavior on the level of manoeuvring in traffic. The cars do not possess skeletal or intrinsic movements, but slow traffic simulation has to take place at several levels. The first level of motion (FLM) consists of the primary motions such as taking a step, taking a turn, or getting off a bike. The behavior of traffic participants in the virtual world forms the second level of motion (SLM). This level is actually a motion planning system that requires information about the environment database. This paper only addresses the problem of creating the first level of motion. The existing second level of motion in the TRC driving simulator will be used or adapted to handle slow traffic participants as well. Only the interface between level two and level one describing how the simulated persons will be controlled has been implemented.

2 System Design

The task of controlling and moving ViPs can be separated into motion specification and motion generation [MC90, BPW93, Ger89]. Motion generation can be subdivided into internal and external motion generation. A conceptual overview of the system's subdivision into a number of logical parts or modules is given in Fig. 2. The motion specification (MS) and motion generation (MG) modules form the core of the system. Other modules are the ViP database and the graphical mapping of the model.

2.1 Skeleton model

A skeleton-like structure can be used both for motion definition as well as the construction of the graphical model. Motions will be defined on the skeleton, and therefore it must be possible to transform postures of the skeleton to postures of the graphical model [BPW93, BBZ91, MTT85, MTT90]. The problem of skeleton joints rotating to unrealistic positions can be solved by constraining each degree of freedom.

2.2 Graphical modeling

The main problem of graphical modeling is to create naturally looking virtual people that can be rendered efficiently. The graphical model constructed from suitable graphical primitives will be wrapped around the skeleton structure. This is called *mapping* the model on the skeleton. Since in the simulator, the ViPs will only be visible from the outside, it suffices to display only the surface of each ViP.

2.3 Motion Specification

The MS module contains rules describing sequences of motions to be generated by the MG module. It is the interface between the user controlling the ViPs and the database of available motions, translating high level commands (HL commands) issued by the user to low level commands (LL commands) of the system. The LL commands are interpreted by the MG module as elementary motion instructions. The ViP database is consulted on the current parameters of each ViP, such as position and orientation in the world, personification parameters, skeleton, etc.

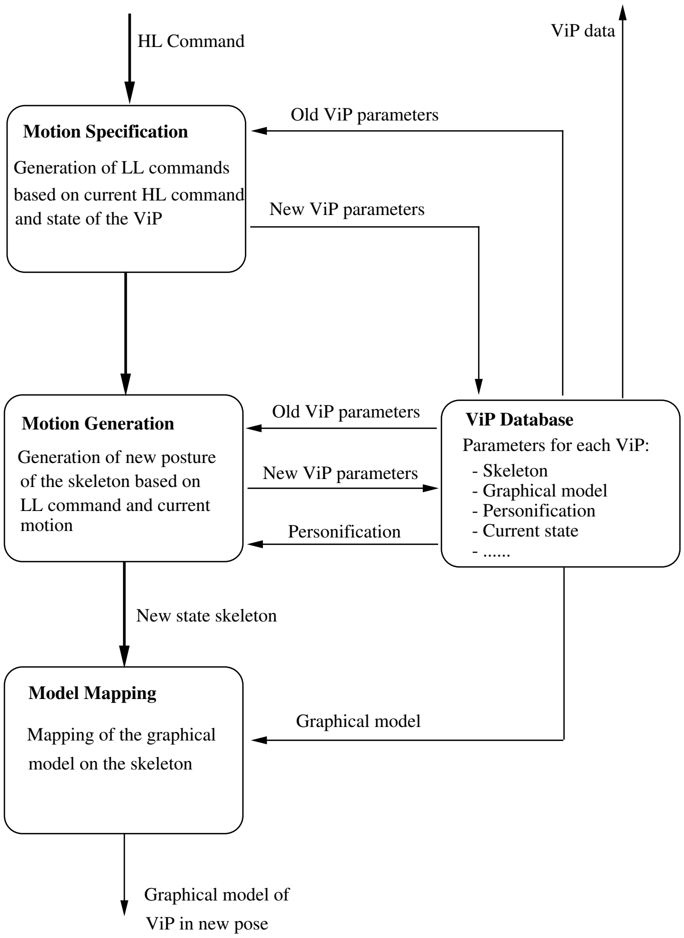

Figure 2: *Conceptual view of the system, showing logical modules with data flow indicated by arrows.*

2.3.1 Motion Specification Graph

An obvious logical structure for specifying motion is a graph structure, in this case a finite state machine, cf. Fig. 3. Nodes in the graph are basic motion states of the ViP. The current node indicates the LL command that is sent to the MG module. Transitions from one basic motion to another correspond to traversals to neighboring states, as indicated by arrows. Transitions have HL command labels or expression labels attached. A transition can only be made if the currently active HL command corresponds to the label, or the attached expression evaluates to true. A default transition is made when no other transitions are valid. The initial state is the state which has an incoming arrow without source state.

2.3.2 Processing HL Commands in Time

An important problem is the delay between the moment that the HL command enters the system and the moment it can be obeyed. This implies that HL commands should be issued by the SLM module before the desired motion has to start, requiring that the MS and/or MG module must be able to predict how long motions will take. Therefore, HL commands will have a parameter,

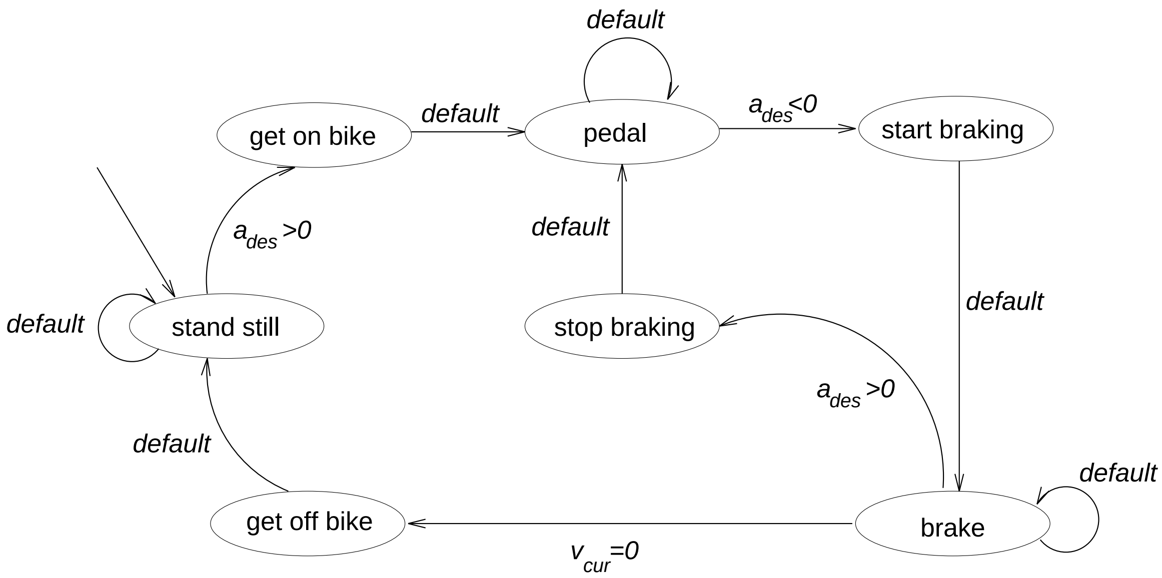

Figure 3: *Finite State Machine for motion specification of the bicyclist. a_{des} is the desired acceleration, v_{cur} the current velocity.*

which specifies at what distance from the current location the desired motion must start. A HL command sent by the SLM module can be obeyed completely if the distance specified in the command is larger than the distance it will take to switch to the specified motion, so that the command arrives 'in time'. The FLM cannot guarantee that HL commands will be processed in time. The higher levels of motion need to handle the consequences of HL commands that haven't been obeyed in time.

2.4 Motion Generation

The MG module determines the new state of the skeleton depending on the current parameters and the LL command to be executed. The new state is returned to the ViP database and also sent to the mapping module. This module maps the graphical model of the ViP onto the skeleton, creating a new posture of the ViP for display. For each frame a LL command is issued, a motion is generated, the ViP database is updated, and the posture of the model is determined. In contrast to LL commands, HL commands are not generated every frame. The SLM has to monitor the behavior of each ViP to guide the ViP through the traffic environment. In Fig. 2, this is indicated by the arrow extending out of the ViP database module pointing upwards.

2.4.1 Internal Motion Generation

A number of methods for generating internal motions have been examined to see whether they would meet the requirements of real time performance, interactivity, naturalness and personification of motion, and the capability of simulating both pedestrians and bicyclists. Among these methods we mention Keyframe Animation, Kinematics, Dynamics, and Motion Capture. In view of the requirements formulated above, we have decided to use motion capture to create a library of basic internal motions. Using these motions and additional algorithms to create new motions by interpolating basic motions, all required motions can be generated in real time.

2.4.2 External Motion Generation

External Motion Generation (EMG) must ensure natural motion of each ViP through the world. The MS module has to ensure that the internal and external MG modules create 'compatible' motion. We limit ourselves here to a discussion of external motions of bicyclists.

For *straight-line motion* of the bicyclist, there are three motion states: starting, moving (accelerating or decelerating) and stopping. A number of parameters are used corresponding to speed and acceleration. Also, depending on the forces slowing the bike down or speeding the bike up, the ViP has to push more or less hard on the pedals. For this purpose we introduce an effort parameter E.

The EMG module uses input parameters like current velocity, desired velocity after a given distance, and personification parameters. From this, it computes output parameters like possible acceleration, the required effort E and required special case of basic motion, using certain expressions based on physical principles like force, mass and acceleration. (The EMG module doesn't use forces directly, that would be tantamount to using *dynamics* for motion generation, which is not the case in our approach.)

The simplest extension of straight-line motion is moving along a *curve* in the horizontal ground plane. In the case of the traffic environment (consisting of straight roads, curved roads and intersections) all that is needed are straight line segments and circle arcs. Therefore we will add circular turns to the motion repertoire.

Two HL commands suffice to specify all 2D motions of the bicyclist:

- Speed $= v_{goal}$ (m/s) after d (m)
- Radius $= r$ (m) after d (m)

A left turn has a positive radius, a right turn has a negative radius. Motion along a straight line is specified by setting the radius to zero. When riding along circles the bicycle as well as the posture of the ViP changes from upright towards a slanting position. The 'banking angle', which is the slant angle with respect to the vertical position, has to be equal to the stable banking angle which can be computed from the physics of centripetal motion. When exiting the turn, the bicyclist has to stop banking and therefore is pushed into an upright position.

3 Prototype System

The motion specification module, motion generation module, and ViP database shown in Fig. 2 have been implemented in a prototype system to evaluate the performance of ViPs before they will be integrated in the TRC driving simulator. In developing the prototype system additional tools were used, such as the PRIMAS motion capture system [vK94] (cf. Section 3.3), Wavefront Kinemation human modeling and animation software, Wavefront Model 3D object modeler, and IRIS Performer, 'a high performance multiprocessing toolkit for real-time 3D graphics' [RH94].

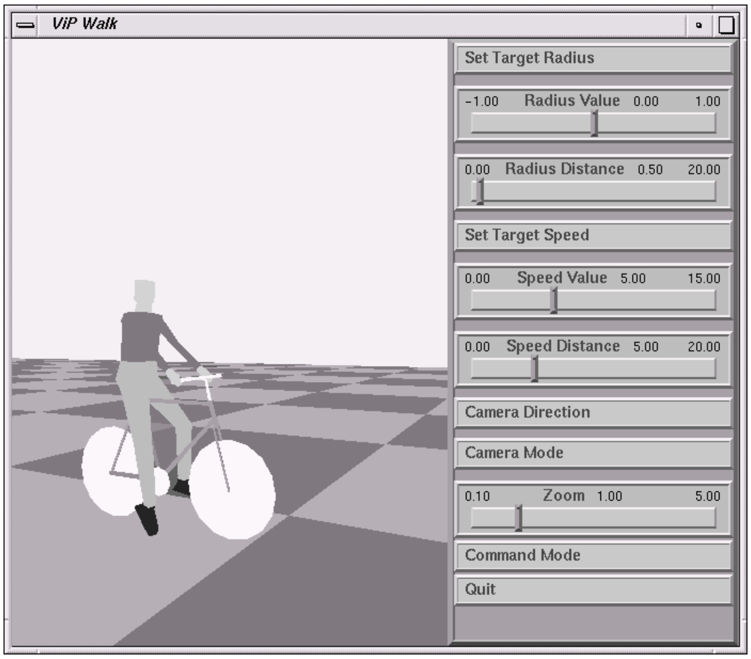

Figure 4: *Screenshot of the prototype system showing a single frame and the user interface.*

The program was implemented on a Silicon Graphics Indy in C++, with IRIS Performer to handle the interface functions and graphics. A screenshot of the graphical control interface is shown in Fig. 4. In the current implementation, the graphical mapping module shown in Fig. 2 is absent. Basic motions are not defined in terms of a skeleton, but directly in terms of the graphical model. The captured motions defined on the skeleton are transformed in a preprocessing stage to basic motions defined on the graphical model using the Kinemation software.

3.1 Graphical Modeling

3D graphics is handled by IRIS Performer. We used only triangle primitives, which can be rendered very efficiently by the SGI graphics hardware. A graphical model of a human person has been created with Wavefront Model, consisting of 224 vertices and 280 shaded polygons. Texture mapping is absent at the moment. A simple bike has also been designed with Wavefront Model, consisting of 236 vertices and 178 polygons.

3.2 Motion Specification

Specification of motion in the prototype system consists of two parts: the user interface and the state machine, as described in Section 2.3. Personification commands have not yet been included.

The user interface of the prototype system is shown in Fig. 4. There are buttons to control the camera mode (fixed or tracking) and camera position (side view or top view). Sliders control parameter values, such as desired speed or radius, which trigger the appropriate HL commands.

3.3 Motion Generation

The basic motions used in the prototype system have been obtained using the PRIMAS motion capture system developed at the Delft University of Technology, The Netherlands [vK94]. The system works with reflectors (markers) attached to the body of the test actor. Positions of the markers are recorded by several cameras.

A skeleton consisting of 14 joints was created based on the marker positions of the captured motions and animated using key frames or by importing captured motion. The graphical model was imported in Wavefront Kinemation and attached to the skeleton. External motion generation uses the steps discussed in Section 2.4.2.

3.4 Performance results

3.4.1 Real Time Performance

In the prototype system, the major bottleneck in the simulation loop is the drawing process. The system has been tested on three SGI platforms.

- INDY workstation with a MIPS R4000 processor and software GL implementation. When the window is very small (80 by 60 pixels), the simulation runs at the maximum frame rate of 60 fps. Full-screen simulation (1280 by 1024 pixels) runs at a frame rate of about 2 fps.

- SKYWRITER equipped with four MIPS R3000 processors and a dual VGXT graphics hardware pipeline. Full screen simulation (1280 by 1024 pixels) runs at 30 fps. When the window is reduced to a quarter of the screen (640 by 512 pixels), the frame rate increases to 60 fps. The pixel-fill rate of the VGXT pipeline is the main cause of lower frame rates at high resolution.

- ONYX equipped with four MIPS R4400 processors and Reality Engine^2 graphics hardware. This provides 60 fps full screen animation (1280 by 1024 pixels) running on one processor. The combination of full screen anti-aliasing and 60 Hertz update results in very smooth motion.

3.4.2 Interactive Control

The standard control method of specifying a HL command with a distance that describes when the command must be obeyed may be useful for the second level of motion (SLM) module in the traffic simulator environment, but it isn't a suitable method for the user controlling ViPs via the graphical user interface of the prototype system. Specifying commands that are directly executed provides the user with a more intuitive control.

3.4.3 Natural Motion

Because motion capture is used, internal motion generation is realistic, although improvements in captured motions and subsequent key frame animation are still possible. Not all current motions connect seamlessly. Also, when the bicyclist is cycling fast, a transition from a left to a right turn or vice versa results in an unnatural rapid change of banking angle.

4 Summary and conclusions

From an evaluation of the prototype system discussed in Section 3, we draw the conclusion that real time interactively controlled computer animation of pedestrians and bicyclists is feasible on modern simulation systems. The combination of motion capture and interpolation for internal motion generation with simple curve following algorithms for external motions results in a bicyclist that is flexible enough to travel through a flat traffic environment such as provided by a driving simulator. Implementation of the ViPs in the simulator will require a powerful platform like a multiprocessor Onyx with Reality Engine^2 graphics hardware in order to obtain acceptable frame rates.

Many improvements of the prototype system are possible. One is achieving more graphical realism with less polygons by a more careful design of the models. This should make it feasible to simulate a dozen of pedestrians and bicyclists at an acceptable frame rate. Basic motions should be defined on a skeleton structure instead of directly on the graphical model, reducing the required amount of memory and speeding up the interpolation process. An efficient method has to be developed to map a graphical model on the skeleton. Also Level-Of-Detail (LOD) switching is a desired addition to the system. A method is needed to ensure smooth transitions of motions, even when the basic motions themselves do not connect smoothly. It will be necessary to include more physical properties into the system to limit the number of unnatural reactions. Complete 3D motion freedom will be required in general environments with viaducts with access ramps and sloping roads. Finally, the prototype system has to be integrated in the TRC Driving Simulator, with ViPs controlled by a SLM module which guides the ViPs through the traffic environment.

References

[BBZ91] N. I. Badler, B. A. Barsky, and D. Zeltzer, editors. *Making Them Move: Mechanics, Control, and Animation of Articulated Figures*. Morgan Kaufmann Publishers inc, San Mateo California, 1991.

[BPW93] N. I. Badler, C. B. Phillips, and B. L. Webber. *Virtual Humans and Simulated Agents*. Oxford University Press, New York, NY, 1993.

[Ger89] M. Gerard. *The Computer Animation of Legged Animals: Simulation, Design and Control*. PhD thesis, Ohio State University, 1989.

[MC90] C. L. Morawetz and T. W. Calvert. Goal-directed human animation of multiple movements. *Proceedings of Graphics Interface*, pages 60–67, 1990.

[MTT85] N. Magnenat-Thalmann and D. Thalmann. *Computer Generated Images*. Springer, New York, 1985.

[MTT90] N. Magnenat-Thalmann and D. Thalmann. *Synthetic Actors in Computer Generated Films*. Springer, New York, 1990.

[RH94] J. Rohlf and J. Helman. Iris performer: A high performance multiprocessing toolkit for real-time 3d graphics. In *ACM Siggraph Computer Graphics Proceedings, Annual Conference Series*, page 381, 1994.

[RvDHvW96] Jos B.T.M. Roerdink, Mattijs J.B. van Delden, Andrea J.S. Hin, and Peter C. van Wolffelaar. Real-time computer animation of bicyclists and pedestrians in a driving simulator. Report CS-R9606, Dept. of Computing Science, University of Groningen, the Netherlands, December 1996. Postscript version obtainable at `http://www.cs.rug.nl/~roe/`.

[vD95] M. J. B. van Delden. Real time computer animation of interactively controlled bicyclists and pedestrians. Master's thesis, Dept. of Computing Science, University of Groningen, the Netherlands, December 1995.

[vK94] J. van Kasteren. Primas zorgt voor natuurlijker beweging in cartoons. *Delft Integraal (in Dutch)*, 4, 1994.

[vWvW95] Peter C. van Wolffelaar and Wim van Winsum. Traffic modelling and driving simulation - an integrated approach. In *Proceeding of the Driving Simulator Conference DSC '95, Sophia Antipolis, France, September 12-13, 1995*. Neuf Associes, Teknea, 1995.