Towards Traffic Generation with Individual Driver Behavior Model Based Vehicles

Benoit Lacroix [1,2], Philippe Mathieu [2], Vincent Rouelle [1], Julien Chaplier [3], Gilles Gallée [3] and Andras Kemeny [1]

[1] RENAULT, Technical Center for Simulation
Technocentre
1 avenue du Golf
78288 Guyancourt Cedex, France

e-mail: {benoit.lacroix, vincent.rouelle, andras.kemeny}@renault.com

[2] LIFL CNRS UMR 8022
University of Sciences and Technologies of Lille
Cité Scientifique bat. M3 F-59655
Villeneuve d'Ascq Cedex, France

e-mail: philippe.mathieu@lifl.fr

[3] OKTAL
8, rue de la Ferme
92100 Boulogne, France

e-mail: {julien.chaplier, gilles.gallee}@oktal.fr

Abstract

Providing realistic road traffic in virtual environments is a challenging problem. It implies handling behavioral and physical aspects of autonomous vehicles, so as to immerse the driver in a realistic simulated environment and provide perceptually valid results. SCANeR$^{©}$ II, developed initially by Renault and today by Oktal, integrates a traffic module which has been developed during the last 10 years. As industrial and academic end-users are becoming more and more exigent about traffic realism, new perceptual and behavioral driver-vehicle behavior models will constitute the basis of an enhanced traffic module for a new version of SCANeR$^{©}$.

In this paper we first present the traffic simulation module implemented in SCANeR$^{©}$ II. Existing features of the current perception-decision-action model, providing autonomous vehicles with realistic interactions between each other, as well as intrinsic limitations and difficulties are discussed. The capacity to describe adequately the environment, with geometric, topologic and semantic data, is an important basis for the animation of the autonomous traffic. SCANeR$^{©}$ II currently uses a two-layer data format for the description of the environment, which is based on a discontinuous logical description mapped on a continuous 3D graphical database.

We present here a new unified data format with a new logical description. We show how the integration of this new data format increases the realism of the behavior of the autonomous vehicles, by reducing conflicts that could appear between graphical and logical layers. It provides also a solid basis to major improvements of the current traffic module, like the introduction of realistic behavioral parameters in the decision model, allowing to take into account various perceptual variables, such as Time To Collision or Time to Lane Crossing, in the execution of the different vehicles maneuvers.

Introduction

Providing realistic road traffic is a challenging problem. It involves driving strategies, convincing trajectories realism, driver's behaviors and decisions, graphics environment. Behavioral realism, one of the main concerns in driving simulators, is also hard to procure, as almost everyone already has a driving experience: tolerance level is low, and each driver has its own judgment on the matter.

As traffic management relies upon the environment's representation, perceiving the world is a key issue in driving simulators. In this paper we address some issues linked to this topic: providing a representation and correlated data format suited to optimum traffic interpretation. Intrinsic improvements brought by the new format as well as assets for further traffic ones are also developed.

We will first outline traffic and environment descriptions currently used in this field, then describe the traffic simulation module and data format which have been implemented in SCANeR© II traffic simulation module. A new unified data format, which solves most of the former encountered issues, will also be presented. This new format provides a solid basis for the next-generation traffic module, which will be described in the last part.

Related work

Traffic modeling

Traffic simulation can be approached in several ways, depending on the requested level of detail. However, when dealing with a driving simulator, only a microscopic representation is suited: the vehicles evolving around the interactive one should have a convincing behavior, which macroscopic or mesoscopic models can not provide.

Various traffic management models have been developed during the last fifteen years. In the ARCHISIM project (Espié, Saad, Schnetzler, Bourlier, & Djemane, 1994) the decision model is based on rules producing intentions (suppression of interactions, adaptation to the context…), and modes associated to these intentions (overtake, change lane…). Only one mode is selected after the decision process, and the associated procedure executed. In Hank (Wang, Kearney, Cremer, & Willemsen, 2005) Hierarchical Concurrent State Machines are used. They introduce hierarchy (each automaton state may itself be an automaton) and parallelism (different states may be simultaneously active, which implies using an algorithm to solve potential conflicts) into the model. Compared to classical automatons this approach enhances modularity and behavior producing. VTISim (Olstam, 2005) includes a microscopic model composed of sub-models for driving behavior, and a mesoscopic model handling vehicles not present in the driver's neighborhood (inner and outer regions are defined, providing a criteria for model switch). This reduces the computer load and allows focusing on the most meaningful vehicles.

Whatever the model used, vehicles perceptions and displacements strongly relie upon the way the environment is described.

The environment: what is needed and why?

Like real drivers, autonomous vehicles involved in simulations use the clues present in the environment to move through it. But when real drivers have no other choice but to depend on their visual capacities and training, artificial entities have more means of perceiving the environment.

The first one is to interpret the environment by directly analyzing the graphical scene, like humans would do. Environment knowledge is extracted from the 3D scene, and interpreted in real time. Such computational vision methods, explored by example in Animat vision (Terzopoulos & Rabie, 1997), involve complex vision algorithms. They are an adequate solution for studying how drivers visually analyze their environment, but are not adapted to traffic simulation.

Another more common approach is to compute the traffic vehicle strategy from a logical description of the database. A multi-layer database – typically two layers, a graphical 3D one and a logical one – constitutes the reasoning basis. This approach is used in most driving simulators (Willemsen, Kearney, & Wang, 2003; Carles & Espié, 1999), because it provides the agents with a controlled semantic vision of their environment (Donikian, 1997). Many complex issues of environment's driver interpretation may be avoided this way.

Standardization approaches have been brought recently, with for example OpenDRIVE® (Dupuis & Grezlikowski, 2006), which is intended to allow logical data exchange between simulation software.

The current traffic simulation module

The Renault/Oktal's SCANeR© II driving simulation software is based on a functional modules architecture using a common communication network (Champion, Mandiau, Kolski, Heidet, & Kemeny, 1999). The main modules are the visual, the traffic, the scenario control, the motion platform control and the supervisor, but handling of data recording or management of video cameras are also available. The traffic module is independent, and can be distributed across different computers to achieve load balancing. In this case, each traffic module manages a subset of vehicles, interacting with its other instances through the network. When one or more real drivers are attending the same simulation, they are taken into account by mapping each dynamic model's data to dedicated vehicles in the traffic.

The decision model

The autonomous vehicles use a classical perception-decision-action architecture as reasoning basis. Their behavior can be adjusted through various parameters, both physical (maximal acceleration and deceleration, weight, maximal steering angle…) and pseudo-psychological (overtaking risk, safety distance, signalization observance…).

First, the perception model identifies the various elements which may interact with the vehicle: roads, lanes, other vehicles, road signs and pedestrians. The detection range differs depending on the vehicle's speed, being computed for a fixed number of seconds. Elements encountered on the path road of the vehicle and elements on their way to the next encountered intersections are computed using the curvilinear distance to the vehicle; short-range and pedestrian detection use cartesian coordinates.

The decision model is built on three levels: strategic, tactical, and operational. The strategic level plans the itinerary. A vehicle may follow a scenario-defined itinerary, or choose its next direction randomly at runtime. No dynamic planning is performed, as it is not usually required in our kind of applications. The tactical level selects the next maneuver to be executed. It uses a finite state automaton to switch between possible actions: stop, drive on, change lane and overtake. Transition switches are computed using scoring functions. Any state can also be requested through scenario rules: in this case, the switch is mandatory. Then the operational level executes the maneuver, by computing acceleration and wheel angle resulting from the choices made in tactical level.

At the action level, vehicle's next position and orientation are computed. To achieve smooth displacements, a spline between the actual position and the goal is used: the next position will be a point of this curve, which is moved to take the speed and wheel angle into account. This flexibility allows each vehicle to create its own trajectory, taking the inner side of a curve, for example.

Road description

In the former versions of SCANeR$^{©}$ II the reasoning was based on a logical description of the environment, the Road Network and Signs (RNS), used by the autonomous vehicles to drive in the scene. A second layer, the Road Surface (RS), was dedicated to the road surface description used by the tire's dynamic model for the interactive vehicle.

The RNS defines the representation of all the different roads in the database. Except for the road surface itself, all the network parameters are specified here: road nature, road width, number of lanes, lane width and driving direction. The signalization is also present in the RNS, and can be placed anywhere along the path. The road path is modeled as a set of points sampled from the central line.

As for the RS, Bezier patches have been implemented to describe the road. They provide the normal vector surface for any point on a road, and ensure a continuous derivative between adjacent patches. This technique makes the surface smoother than the graphics polygon description. Road surface parameters are included: adherence, spatial noise (smooth to bumpy, corresponding to new asphalt to cobblestones), nature (asphalt, snow...), and road type (normal, sidewalk, obstacle...). It does not include any logical element.

This two layers data separation had been selected to be able to achieve the simultaneous management of more than one hundred vehicles: using the continuous description of the RS to compute vehicles displacement was much too costly. With the availability of increased computing power this kind of limitations has been substantially reduced and such compromises becomes no longer necessary.

Improving the logical description to act upon traffic realism

Ways of improvement

The Road Network and Signs has nevertheless some intrinsic limitations, most of them caused by the discontinuous description of the road. This lead to two kinds of issues: graphical glitches and abnormal behaviors.

When driving on slopes or on banked roads, graphical glitches can appear. Slope placement approximations are induced by the insufficient density of points describing the road: the vertical position of the vehicles is calculated for the middle of the rear axle. When the slope is too high, the front wheels graphically appear inside the road (Figure 1.a). As for the banking, the RNS does not include this information, and therefore may create inconsistencies with the graphical layer. Another problem concerns the crossing of the sidewalks by the pedestrians. Indeed, they sometimes make a step inside the pavement (Figure 1.b), for the same reason as vehicles above: their logical representation does not perfectly reflect the graphical one, and they perceive too late the elevation caused by the sidewalk.

Figure 1: The discontinuous logical description of the road lead to graphical glitches.

Some abnormal behaviors linked to the data format also occur. First, at the top or bottom of elevations and in curves, vehicles sometimes slow down abruptly, without any visible reason. The density of the points in the RNS description is a function of the road curvature: this number is thus more important in bends and elevations. But the computation of the next vehicle position is based only on the three road's nearest points. When these points are very close, like in the above cases, the vehicle acts as if the radius of curvature was very high, and slows down accordingly.

Finally, the various files corresponding to the same base are sometimes hard to manage for the end users. Consistency problems between them can also appear, and are hard to detect.

Aim and description

A new version of the road network has thus been developed to address two main issues: correcting the various troubles encountered with the RNS, and being scalable and flexible enough to allow further improvements.

This new data description format, the Road Network Description (RND), merges the Road Network and Signs logical data layer with the continuous aspect of the Road Surface layer. The unification of the different bases means that the edition, the dynamic model and the traffic all proceed on the same data. Almost all elements are included in the format, except for some which are still directly added at a graphical customization stage (manhole covers for example). In addition, RND is compatible with the OpenDRIVE® standard.

The network is constituted of tracks and nodes. Nodes have an *(x, y, z)* position, and one or more tracks are connected to them. Tracks contain the road axis, as a function linking two nodes. Three function of the curvilinear abscissa describe the different properties of the road: the first expresses the cartesian coordinates ($F(s) = x_1, y_1$), the second the elevation ($G(s) = z_1$), and the third the banking. Road axis trajectories have to match a given set of mathematical functions. In the *(x, y)* plot, these functions include straight lines, clothoids and arc of circle. Keeping a C^2 continuity implies to intercalate clothoids between straight lines and arc of circles. In the *(s, z)* plot, roads are collections of parabola and segments. There is no vertical C^2 continuity.

After this step the pavement is characterized through a transverse definition (Figure 2.a). This transverse definition includes the road profile, split in three layers: the profile layer (which allows using different properties on the same track), the road surface layer (the adherence is stored here, for example, see Figure 2.b), and the traffic layer.

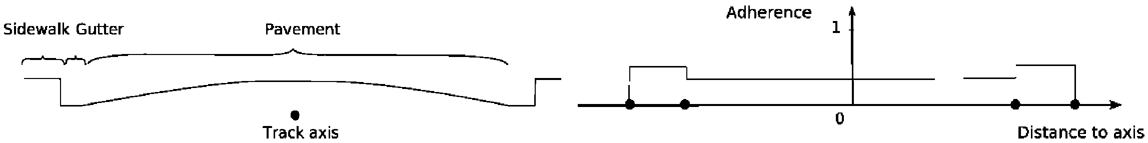

Figure 2: (a) Transverse section of the road. (b) A property of the road surface layer: adherence function of distance to the axis.

Intersections require specific information, organized around a node. An intersection area is defined: the proximity of the intersection is defined by a distance parameter stored in each connected track. Intersections supply additional data: priority levels of the tracks, available in and out ways, and allowed connections between them. Such architecture allows for example to forbid U-turns for all or specific vehicles. Predetermined path between ways may be included, so as to ease vehicles displacement in intersections.

Diverse elements can be locally added to the network. These local information include the vertical signalization (signs, lights…), surface elements (local perturbation of the pavement, stop lines…), beacons and obstacles. Some higher level data structures, roads and itineraries, are also available. Roads group tracks, and each track may be used in different roads definitions. Itineraries can also be created. These structures, made available in the traffic, can then be exploited in the software logic.

Heavily scalable, RND's format has two main assets: the road's transverse definition and the possibility to add user specified data directly inside the description. Some of the network properties can indeed be extended: pavement, tracks, nodes, lines and vehicles. These new data can then be exploited in SCANeR© II through user defined plugins.

What does it allows in the current traffic?

The addition of new roads transverse profiles allows solving the banked roads glitches: required data are now available. All vehicles are provided with the normal vector surface for any point on a road, and each wheel can be individually computed, so as to be placed exactly at the road surface. The mechanism used for the slope case is similar: the function describing the road axis can be used directly, without any sampling side effect, and wheels can then be positioned at the required position.

Figure 3: Example of graphical glitches corrections brought by RND in SCANeR©II

As for the unexpected slowdowns encountered at top or bottom of elevations, the continuous aspect of the road supplied here again the solution. Abnormal behavior was caused by the unexpected result of road curvature computation; this curvature is now directly provided by the road description.

As will be seen below the new format also introduces features allowing the improvement of the traffic realism itself.

Towards a behavior driven model

The current model provides many features, but industrial and academic end-users are becoming more and more demanding for high level traffic realism. Vehicles behavior and individual differences are two elements which have to be improved, taking advantage of the new road format.

Addressed issues

Currently all vehicles (trucks, cars, motorcycles…) behave similarly, using the same model. This model takes both physical and pseudo-psychological parameters into account. Pseudo-psychological ones have a limited influence on the vehicles behavior: overtaking risk parameter affects the automaton's state transition; maximal speed and speed coefficient allow the vehicles to bypass speed limitations; safety time impacts the following distance; observe signs and observe priority parameters are boolean rules regarding signalization. However, once set, these parameters do not evolve, except through scenario rules. For example, a vehicle will always try to reach its maximal desired speed. In a similar way physical parameters are involved in traffic model computation (maximal speed, angle, acceleration, and braking), but external circumstances have no impact on their taking into account.

As for the intersections, no global supervisor is present to handle them specifically, although it is a commonly used method. The vehicles remain totally autonomous, hanging on their perception of the environment. The possibility to use corridors to cross complex intersections brought through the OpenDRIVE® compatibility remains limited: the taking into account of large vehicles for example can hardly be managed this way, as they would need a dedicated corridor, even a different one for each possible vehicle size... But user defined data enable the introduction of traffic specific markers, which can be used to introduce additional control points inside crossroads to improve trajectories.

Behavioral differentiation and informal rules

Several psychological factors are involved in driver's behavior (Dewar, 2002): personality, emotion, motivation and social behavior. Psychological based driver models have been developed (Keskinen, Hatakka, Laapotti, Katila, & Peraho, 2004), but the lack of links between measurable and psychological parameters makes their concrete application difficult.

Psycho-physical experiments conducted on the driving simulators have led to the integration of some of these parameters in SCANeR© II (Time To Collision, Time to Line Crossing…). They are thus available for experimentations, but not exploited in the decision model. Experimental data, describing for example the probabilistic distribution of these parameters values among drivers population are available (Van der Horst & Hogema 1993). They will be used to calibrate the model and introduce the behaviors variety encountered in the real world, like aggressive driving. Intuitive, independent and self-coherent parameters will be available: whatever the selected parameters set, the resulting behavior remains realistic.

Another interesting way to improve vehicles behavior's realism is to enable them to take into account the various rules encountered in the real world (Björklung & Aberg, 2005):

- formal rules (rules of the road),
- informal rules (practices or conventions which can be in contradiction with the formal rules, like not yielding at crossroads or roundabouts),
- design of the road (which is often the origin of informal rules appearance),
- and other drivers behavior (their current behavior as well as the anticipated one).

Every driver obviously does not have the same informal rules sets: consistent aggressive, formal, informal and cautious groups emerge.

Currently either vehicles respect the rules of the road, or they do not. The influence of other vehicles behavior will be taken into account. For example accelerating or decelerating when arriving at a crossroad affects others behavior. Even if a vehicle has the right of way, a vehicle arriving at what it thinks is a “too fast” speed will lead it to yield, so as not to run the risk of a crash. Gap acceptance and related factors will be used to simulate these anticipation behaviors.

Conclusion and future work

We have presented the new road description format developed for the SCANeR© II software. It corrects the graphical glitches caused by the former format (on banked roads and slopes for example), and some abnormal behaviors caused by the distorted vehicles perceptions are also solved. Among future evolutions an interesting approach would be to create links with description formats like LandXML (“LandXML file format”, 2007), a XML file format for civil engineering design and survey measurement data. The flexibility and scalability introduced at the conception through user defined data and various transverse profiles provide a solid basis for further improvements.

This work led to the development of an enhanced traffic model, based on behavioral parameters easily understandable by the end users. Data available from the psychological studies can then be exploited for calibration and variability of driver’s profiles. New behaviors connected to informal rules of the road, taking into account the behavior of the other road users besides pure geometrical data, will also be introduced. They will enhance the perceived behavior of the autonomous vehicles by the drivers using the simulators.

References

Björklung, G. & Aberg, L. (2005), Driver behaviour in intersections: Formal and informal traffic rules. *Transportation Research Part F*, pp. 239-253.

Carles, O. & Espié, S. (1999). Database generation system for road applications. *Proceedings of the Driving Simulation Conference*, pp. 87-103.

Champion, A., Mandiau, R., Kolski, C., Heidet, A. & Kemeny, A. (1999). Traffic generation with the SCANeR© II simulator: towards a multi-agent architecture. *Proceedings of the Driving Simulation Conference*, pp. 311-324.

Dewar, R. E. (2002). Individual Differences. In R. E. Dewar, P. L. Olson (Ed.), *Human Factors in Traffic Safety* (pp. 111-142). Lawyers & Judges Publishing

Donikian, S. (1997). VUEMS: a virtual environment modeling system. *Computer Graphics International*, pp. 84-92.

Dupuis, M. & Grezlikowski, H. (2006). OpenDRIVE® – an open standard for the description of roads in driving simulations. *Proceedings of the Driving Simulation Conference*, pp. 25-36.

Espié, S., Saad, F., Schnetzler, B., Bourlier, F. & Djemane, N. (1994). Microscopic traffic simulation and driver behaviour modelling: the ARCHISIM project. In *Road Safety in Europe and Strategic Highway Research Program*, pp. 22-31.

Van der Horst, R. & Hogema, Je. (1993). Time-to-collision and collision avoidance systems. *Proceedings of the International Cooperation on Theories and Concepts in Traffic Safety workshop*, Salzburg, Austria.

Keskinen, E., Hatakka, M., Laapotti, S., Katila, A., & Peraho, M. (2004). Driver Behaviour as a Hierarchical System. In T. Rothengatter and R. D. Huguenin (Ed.), *Traffic and Transport Psychology* (pp. 9-24). Elsevier.

LandXML file format, http://www.landxml.org/, retrieved on June 11, 2007.

Olstam, J. (2005). Simulation of Rural Road Traffic for Driving Simulators. *Proceedings of the 84th Annual meeting of the Transportation Research Board*, Washington D.C., USA.

Terzopoulos, D., & Rabie, T. (1997). Animat Vision: Active Vision in Artificial Animals. *Videre: Journal of Computer Vision Research, 1*, 2-19.

Wang, H., Kearney, J., Cremer, J., & Willemsen, P. (2005). Steering Behaviors for Autonomous Vehicles in Virtual Environments. *Proceedings of the IEEE Virtual Reality Conference*, pp. 155-162.

Willemsen, P., Kearney, J., & Wang, H. (2003). Ribbon Network for Modeling Navigable Paths of Autonomous Agents in Virtual Urban Environments. *Proceedings of the IEEE Virtual Reality Conference*, pp. 79-86.