

# A Faust Based Driving Simulator Sound Synthesis Engine

Romain Michon<sup>1</sup>, Mishel Johns<sup>2</sup>, Sile O'Modhrain<sup>3</sup>, Nick Gang<sup>1</sup>, Nikhil Gowda<sup>2</sup>, David Sirkin<sup>2</sup>,  
Chris Chafe<sup>1</sup>, Matthew James Wright<sup>1</sup> and Wendy Ju<sup>2</sup>

<sup>1</sup>Center for Computer Research in Music and Acoustics (CCRMA), Stanford University

<sup>2</sup>Center for Design Research (CDR), Stanford University

<sup>3</sup>University of Michigan

rmichon@ccrma.stanford.edu

## ABSTRACT

A driver's awareness while on the road is a critical factor in his or her ability to make decisions to avoid hazards, plan routes and maintain safe travel. Situational awareness is gleaned not only from visual observation of the environment, but also the audible cues the environment provides - police sirens, honking cars, and crosswalk beeps, for instance, alert the driver to events around them.

In our ongoing project on "investigating the influence of audible cues on driver situational awareness", we implemented a custom audio engine that synthesizes in real time the soundscape of our driving simulator and renders it in 3D. This paper describes the implementation of this system, evaluates it and suggests future improvements. We believe that it provides a good example of use of a technology developed by the computer music community outside of this field and that it demonstrates the potential of the use of driving simulators as a music performance venue.

## 1. INTRODUCTION

It is tempting to think that someday, when we have fully autonomous vehicles, we will be able to clamber into our cars and take a nap on the way to wherever we are headed as the vehicle takes over. And yet, consider how we behave today, when we have autonomous humans, such as taxi drivers, take us from point A to point B. Normally, we passengers feel the need to supervise drivers, at least somewhat, to make sure that they understand our intentions, that they are competent drivers, and that they are not lingering or adding unnecessary waypoints along the journey. Given the challenges associated with perceiving the roadway and associated obstacles, synthesizing many streams of data to develop a coherent model, and the inherently open-ended set of things that we encounter on the road every day, it is altogether likely that people will need to maintain some level of awareness over automation in cars for many years to come.

Situation awareness (SA) was defined by Endsley [1] as

Copyright: © 2016 Romain Michon, Mishel Johns, Sile O'Modhrain, Nick Gang, Nikhil Gowda, David Sirkin, Chris Chafe, Matthew James Wright and Wendy Ju et al. This is an open-access article distributed under the terms of the [Creative Commons Attribution 3.0 Unported License](https://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

"the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future." Such awareness of the surroundings and the situation is necessary for drivers of road vehicles, in order to maintain safe travel. Situational awareness is gleaned not only from visual observation of the environment, but also the audible cues the environment provides - police sirens, honking cars, and crosswalk beeps, for instance, alert the driver to events around them.

Prior work compares spatialized auditory feedback methods [2] and describes the use of spatialized audio for navigation cues [3].

In our ongoing project<sup>1</sup> at Stanford University's *Volkswagen Automotive Innovation Lab* (VAIL) on "investigating the influence of audible cues on driver situational awareness", we are trying to test the following hypotheses:

- Selective amplification/modification of explicit signals will increase the driver's situation awareness.
- Audible rendering of explicit events will affect the driver's spatial recall and sense-making of the surrounding events and the vehicle's actions.
- Selective amplification/modification of ambient signals will make the drive more enjoyable for the driver.
- Audible rendering of ambient context will affect spatial recall.
- Possible interaction effect where using sonification becomes distracting and makes the driver less effective in prescribed tasks.

The Stanford driving simulator uses the RTI<sup>2</sup> (*Realtime Technologies Inc.*) software to design and run the simulations in the frame of this study. While this environment provides a comprehensive set of tools to create virtual driving spaces, it has a quite limited synthesis engine that only renders the sound of the simulator car engine, the road and other cars passing by. Additionally, the sound of the simulation is spatialized only in 2D on a simple 5.1 audio system.

<sup>1</sup> This project is the fruit of a collaboration between Stanford University's Center for Computer Research in Music and Acoustics (CCRMA), Center for Design Research (CDR) and Renault Innovation Silicon Valley.

<sup>2</sup> <http://www.simcreator.com/simulators.htm> Accessed: 2016-07-11.

The simulator is equipped with an automated mode that was used to transport participants through a predefined course. During certain sections of the course, experimenters were required to take over control using a secondary steering wheel in a location not visible to participants. This is known as a “Wizard of Oz” control scenario, and it was used as a more flexible alternative to programming every autonomous movement in the simulation. To better present auditory cues from outside the car, the front windows were left cracked open about 10cm. The rear windows were fully closed.

The simulated course used for our study begins on a parking lot where a series of events happen sequentially around us: a child shouts and runs right next to our car, a bicyclist rings a bell and crosses the road in front of us, a dog barks in reaction, and a garbage truck is heard in the distance (see Figure 1). Synthesizing and rendering all these sonic events required a much more versatile audio engine than the one at our disposal. We needed to be able to easily add any sound source to the simulation and to spatialize them in the three dimensional space of the simulator. We also wanted to improve the quality of the generated sounds by using physical models instead of samples when this was possible.



**Figure 1.** Screenshot of the beginning of the simulation used for our study.

While advanced audio engines for car simulators have already been implemented and described in previous work [4, 5], we decided to implement our own from scratch in order to custom tailor it to our needs and to make it open source.

In this paper, after providing a detailed description of the implementation of our custom audio engine, we evaluate it and provide suggestions for its improvement.

## 2. HARDWARE

### 2.1 Driving Simulator

Simulators are cost-effective ways to evaluate performance on tasks that are too dangerous or improbable in real life. They possess advantages over on-road vehicle testing in that the experimenter has better control over the vehicle surroundings and situations, and can provide for easier data collection [6].

The Stanford Driving Simulator is a high fidelity full-car automotive simulator installed by *Realtime Technologies, Inc.* The car is a 2010 *Toyota Avalon*, with connections to the onboard CAN, a set of digital and analog multipurpose inputs and outputs, and a TCP/IP interface. A mixed WAGO analog and digital I/O System is used by the host

computer to send and receive signals from the car. Twenty-four digital I/O ports and 16 analog I/O ports can be used to connect the system to other devices. Steering feedback is provided by a control loading steering motor that returns torque to the wheel based on vehicle speed, suspension parameters and steering angles. Brakes use the OE hydraulic brake system with a vacuum pump replacing the vacuum normally supplied by the engine.

The visual system consists of a twenty-four foot diameter 260° cylindrical screen. Images from five high-resolution projectors are blended together form a seamless display. LCD inserts for the side mirrors and a rear projector reflected on a rear-view mirror complete the visual immersion in the simulated environment. The side mirror adjust buttons control the eye-points for these LCD inserts.

The Stanford Driving Simulator has an autonomous mode that replicates the experience of being in a self-driving vehicle. The system uses the simulations own traffic control algorithms to calculate the parameters to drive the car. The steering wheel moves on its own when the automated mode turns the car. The gas and brake pedals, however, do not move when the car is automated. The mode (autonomous or manual) of the car is displayed on an icon in the center of the instrument panel.

The pitching of the visuals due to acceleration and braking, as well as tight curves on the simulated roadway do tend to cause discomfort and simulator sickness in a small fraction of our participants. This is probably because of the immersive visual experience that makes the differences between the inputs to the visual and the vestibular systems more noticeable. The solutions that seem to help reduce nausea are to avoid tight turns in the course, enable the air conditioning system in the cabin with cold air blowing towards the driver, and to keep ginger candy and cold water available for participants. In this study, there were no sessions that needed to be halted due to simulator sickness.



**Figure 2.** The Stanford Driving Simulator.

### 2.2 Audio System

The sound system of the Stanford driving simulator consists of a simple sphere of speakers placed on three different levels:

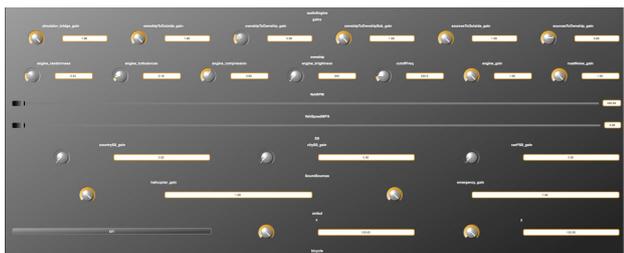
- a ring of four speakers around the car,
- a ring of four speakers right above the roof of the car,

- a center speaker on the ceiling of the simulator room.

In addition to that, 4 speakers in the passenger’s compartment of the car (two on each side of the front and back seats) and a large speaker mounted on the driver’s seat to transduce road vibrations can be accessed. Speakers are controlled using a set of two audio interfaces<sup>3</sup> connected together using an ADAT LightPipe and plugged to a *Mac Mini*. The *Mac Mini* is connected to the simulator computer cluster via ethernet and runs the custom audio engine (see Figure 4). Having a dedicated computer to carry out this task made prototyping easier.

### 3. SOFTWARE

The sound synthesis engine of the Stanford driving simulator was implemented with the FAUST programming language [7]<sup>4</sup> and is compiled as a standalone C++ application with a Qt<sup>5</sup> user interface (see Figure 3). Each of its parameters can be controlled using OSC [8]<sup>6</sup> allowing the simulator software to interact in real time with the sound synthesis engine by sending raw UDP messages. Those messages are converted to OSC using a small program that we wrote in C++ and which runs independent of the other process (see Figure 4).



**Figure 3.** Screenshot of the FAUST generated user interface of the driving simulator sound synthesis engine.

All the generators of the sound synthesis engine render sounds in 3D and access the different speakers of the simulator independently. The signals sent to the speakers of the passenger’s compartment are processed by fourth-order lowpass filters with a cut-off frequency of 500 Hz to reproduce the effect of the car’s shell on interior sounds coming from the soundscape when windows are closed. Similarly, the signal of the driver’s seat speaker is lowpass filtered at 90Hz.

A reverberator based on a FAUST implementation of *freeverb*<sup>7</sup> is used to model the effect of passing through closed structures like tunnels, etc. on the soundscape (see Figure 4).

The synthesized car engine sound is based on a physical model made out of an aperiodic pulsetrain generator fed through a series of filters and effects [9]. This kind of model was designed to simulate the sound of race cars for

video games and was not well adapted to our use. Therefore we had to spend a fair amount of time tuning the model so that it would sound more like a small four cylinder engine of a *Toyota Avalon*. Since the RPM of the engine is controlled directly by the simulator software, we didn’t have to implement features such as gear shifts, engine torque and resistance, etc.

The road noise is synthesized using a noise generator processed by a lowpass filter whose cut-off frequency and gain are adjusted in function of the speed of the car (see Figure 5). The faster the car, the greater the cut-off frequency and the gain.

An algorithm similar to the one presented in Figure 5 is used to synthesize the sound of other cars present in the simulation. The only difference comes from the fact that the cut-off frequency and the gain of the lowpass filter are calculated as a function of the distance between our car and the other cars.

The different sound sources of the simulation can be moved in the 3D space of the simulator by using a spatialization function (see Figure 6). The simulator software provides real-time cartesian coordinate of the different sound sources of the simulation relative to the position of our car. The audio engine converts those coordinates to a set of distances, angles (azimuths) and elevations to carry out *Vector Based Amplitude Panning* (VBAP) [10]. The first-derivative of distance drives a Doppler effect. Any sound source can be easily added to the audio engine by providing an audio sample that can be looped and sent to the spatialization function.

The different elements of the audio engine were condensed into a Faust library<sup>8</sup> in order to easily customize it and reuse it.

### 4. EVALUATION AND FUTURE DEVELOPMENTS

Faust’s compactness enabled us to build our audio engine in less than 400 lines of code and the efficiency of the generated C++ code only requires 12% of the CPU of our *Mac Mini* when simultaneously calculating 28 sound sources, the car engine physical model, the sound of the road, a reverb and a series of filters (see Figure 4).

Neither is the “typical roaring sound” synthesized when the windows of the car are open and the car is moving. More transient sonic events could be added especially when driving through a city to improve the simulation’s realism. Investigation of further details of the car engine sound model remains a top priority in our future work. We will eventually integrate computation of the entire sound system natively in the RTI simulator thereby eliminating the need for a separate sound computer.

### 5. CONCLUSIONS

We have built a versatile, fully customizable and efficient synthesis engine compatible with the RTI software used in the Stanford driving simulator. It is implemented in the FAUST programming language and it can be easily interfaced

<sup>3</sup> *Focusrite Scarlett 18i20*.

<sup>4</sup> <http://faust.gamed.fr> Accessed: 2016-07-11.

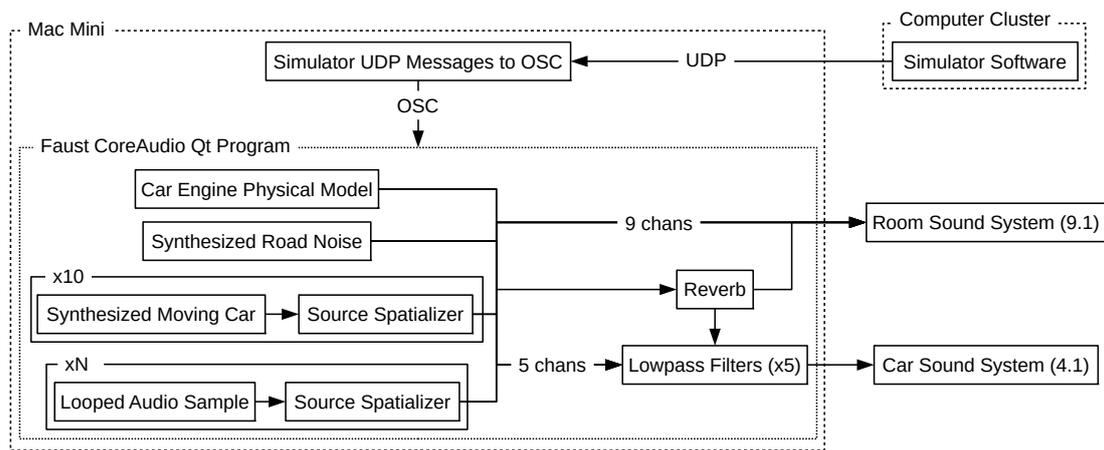
<sup>5</sup> <http://qt.io> Accessed: 2016-07-11.

<sup>6</sup> *Open Sound Control*.

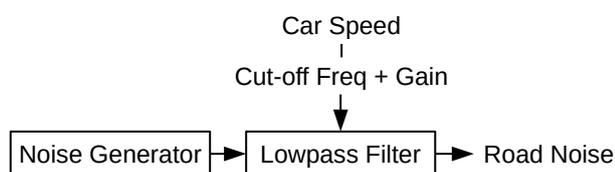
<sup>7</sup> <https://ccrma.stanford.edu/~jos/pasp/Freeverb.html> Accessed: 2016-07-11.

<sup>8</sup>

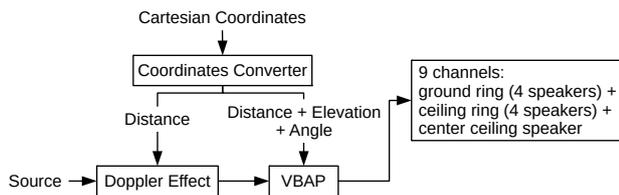
<https://ccrma.stanford.edu/~rmichon/faustDrivingSimulator> Accessed: 2016-07-11.



**Figure 4.** Overview of the implementation of the driving simulator synthesis engine.



**Figure 5.** Overview of the implementation of the road noise generator.



**Figure 6.** Overview of the implementation of the source spatializer.

with any software thanks to its OSC interface. We hope that it will help improve the driver’s experience and provide an environment to design more advanced simulations and audio-related studies useful in the development of autonomous driving.

More generally, we believe that driving simulators provide a standardized complete environment to design interactive art installations. Investigating this unexploited potential should be greatly facilitated by our custom audio engine that provides an ideal prototyping environment to play with that kind of ideas.

### Acknowledgments

We thank *Renault Innovation Silicon Valley* for funding this project as well *moForte Inc.* for allowing us to use their car engine physical model algorithm.

### 6. REFERENCES

- [1] M. Endsley, “Toward a theory of situation awareness in dynamic systems,” *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 37, no. 1, pp. 32–64, 1995.
- [2] D. Beattie, L. Baillie, M. Halvey, and R. McCall, “What’s around the corner? enhancing driver awareness in autonomous vehicles via in-vehicle spatial auditory displays,” in *Proceedings of the 8th Nordic Conference on Human-Computer Interaction*, New York, NY, 2014, pp. 189–198.
- [3] M. Cohen, O. N. N. Fernando, T. Nagai, and K. Shimizu, “Back-seat driver: Spatial sound for vehicular way-finding and situation awareness,” in *Proceedings of the Japan-China Joint Workshop on Frontier of Computer Science and Technology (FCST’06)*, November 2006.
- [4] M. Blommer and J. Greenberg, “Realistic 3d sound simulation in the virtex driving simulator,” in *Proceedings of ASME DSC North America*, Dearborn, MI, October 2003.
- [5] T. Funkhouser, N. Tsingos, and J.-M. Jot, “Survey of methods for modeling sound propagation in interactive virtual environment systems,” Technical Report, 2003.
- [6] J. Gruening, J. Bernard, C. Clover, C. Clover, and K. Hoffmeister, “Driving simulation,” *Vehicule Dynamic Simulation*, 1998.
- [7] Y. Orlarey, D. Fober, and S. Letz, “An algebra for block diagram languages,” in *Proceedings of the International Computer Music Conference (ICMC-02)*, Gothenburg, Sweden, 2002.
- [8] M. Wright and A. Freed, “Open Sound Control: A new protocol for communicating with sound synthesizers,” in *Proceedings of the International Computer Music Conference*, Thessaloniki, Greece, 1997.

- [9] K. Cascone, D. T. Petkevich, G. P. Scandalis, T. S. Stilson, K. F. Taylor, and S. A. V. Duyne, "Apparatus and methods for synthesis of internal combustion engine vehicle sounds," US Patent US 6 959 094 B1, 2005.
- [10] V. Pulkki, "Virtual sound source positioning using vector base amplitude panning," *Journal of the Audio Engineering Society*, vol. 45, no. 6, pp. 456–466, June 1997.