

## **Driving Simulation for Crash Avoidance Warning Evaluation**

Pete Tinker, Ronald Azuma, Cheryl Hein, and Mike Daily

Hughes Research Laboratories, USA

96VR041

### **ABSTRACT**

This paper describes the development of a fully-integrated, medium-performance, fixed-base driving simulator for testing and evaluating crash avoidance warning systems. Three major aspects of the ongoing effort are described: our motivation for constructing the simulator, the system architecture, and the crash avoidance warning systems. The paper concludes with an assessment of our progress to date. The physical simulator is a modified automobile cab instrumented to provide driver activity information and in-vehicle warnings that accurately reflect the performance of proposed sensors during defined scenarios. The simulation computing architecture uses a distributed multi-CPU environment to support six out-the-window displays, a head-up and a head-down display, six channels of audio output, and software to control the simulation and produce warnings.

Planned evaluation with a population of subjects in a variety of scenarios will determine the best warning systems for use in prototype vehicles. We describe the hardware and software in place to provide visual, multi-channel audio, and haptic warnings corresponding to the simulated environment and scenario. We also describe a typical scenario and the process leading to manifestation of warnings of an impending collision.

### **1. Introduction**

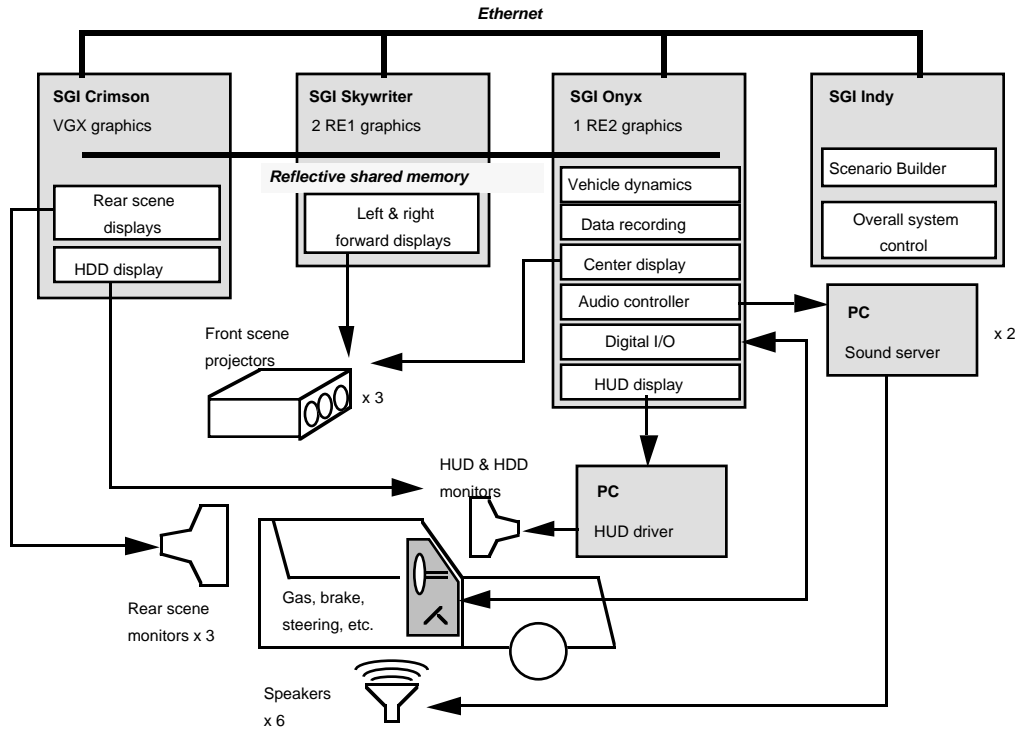
Safety has proven to be a discriminator in marketing automobiles. Airbags, antilock brakes, daylight running lamps and head-up displays are examples of safety devices that have been deployed in the past few years. Future automobiles will feature even more safety systems, including both active and passive sensors for enhanced threat detection and alerting drivers to impending collision situations. The National Highway Transportation Safety Administration (NHTSA) concludes that 70% of rear-end collisions were caused primarily by driver inattention. The Delco Electronics business unit of Hughes Electronics is investigating vehicle-mounted radar sensors that will provide drivers with advance warning of impending collisions, potentially mitigating a large number of collisions caused by driver inattention.

Deploying these sensors-based alerting systems raises a human-interface question: what is the most effective way to present the warnings to the driver? Because driver reaction time is critical, it is important to determine the warnings that are understood most quickly. Furthermore, how does one test and evaluate warnings? Testing and selecting warnings requires placing the driver and vehicle into a situation where a crash is imminent. Due to the obvious danger, our initial evaluation phase will be done in a simulator. Using a driving simulator to test crash avoidance warnings allows exploration of a wide variety of warnings under controlled conditions without risking physical harm to the driver or the test vehicle. Reliable reproduction of the same scenarios across different drivers is another advantage conferred by simulation. Use of appropriate within-subject experimental designs and subject motivating strategies will provide a realistic testing environment while eliminating risks to personal safety.

Many driving simulators exist or are being built. Examples include the Iowa Driving Simulator [0] and others [1]. These simulators vary widely in their purpose and their technical features. The car simulator we use is an updated version of an earlier simulator built by Hughes in 1987. Our goal was not to build a driving simulator that was state-of-the-art in all areas. Instead, taking a driver-centered approach, we designed it to provide the functions required to adequately test driver alerting systems, without expending additional resources on areas that are not crucial to our application. For example, we do not need a motion base because the behavior of the own vehicle is not of high interest; however, the visual field-of-

view and fidelity requirements are high because of the immersive quality of simulation we need to present realistic scenarios.

## 2. System Architecture



**Figure 1 System Architecture**

The complete system (Figure 1) comprises a fixed-base simulator, a front-projected, wide field-of-view toroidal video projection screen, rear view monitors, a rear-projected video projection screen, and a suite of processors to drive the computer simulation. The simulator cab provides a realistic interior environment that also incorporates the prototype warning systems and instrumentation to provide feedback and inputs to the computer simulation. The computer suite provides simulation of preplanned multi-vehicle scenarios, sensor and threat assessment simulation, and rendering of eight visual channels, six audio channels, and vehicle dynamics. The following sections describe the physical simulator, the computational architecture, and the scenario visual database.

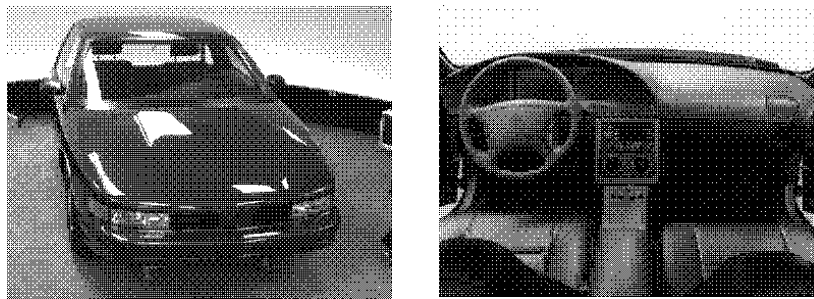
### 2.1 Simulator Cab

The driving simulator cab connects the human driver to the driving simulator. The cab is a specially modified production vehicle and provides a highly realistic environment for the driver. Actual or simulated inputs are produced for the driver in three major classes of sensory modalities: visual, auditory and motion-sensing (tactile, proprioceptive, vestibular). With the remaining simulator components (out-the-window visuals, computational models, and various computers) the cab provides high fidelity sensory cues for all human sensory modes except major motion (lacking primarily vestibular and some proprioceptive cues).

The cab can be thought of as a subsystem in a larger closed-loop feedback system that includes the driver as well as all other simulator components. The cab gathers inputs from the driver and relays them to the simulator computers. Once those computers and their models interpret the inputs and determine appropriate responses, they send various multi-modal outputs to the cab. Thus the illusion that one is driving is completed.

Our simulator cab (Figure 2) is a modified Oldsmobile Cutlass Supreme. The vehicle was cut at the B pillar and the front half was retained for use as the simulator cab. It was further divided into two sections approximately at the firewall, providing modularity for mounting simulation hardware and for ease of transport and storage. Much of the vehicle structure and most non-electronic systems were removed, leaving a lightweight open shell which could be reconfigured as necessary. Significant portions of the metal panels surrounding the driver were replaced with fiberglass to minimize interference with magnetic sensors used in our eye tracking systems.

The cab interior was redesigned to provide an updated, realistic and finished driving cockpit that provides important face validity cues to the driver. The steering wheel, brake pedal and accelerator pedal are instrumented with digital encoders that provide data to close the primary control loop between the driver and the vehicle. The cab provides self-contained cables and wiring for distributing and acquiring digital and analog data, audio and video, and power.



**Figure 2 Cab Exterior and Interior**

The steering column includes a custom built control loader utilizing a torque motor to provide position- and rate-based torque feedback to the driver through a full 2.5 turns of the wheel. The brake pedal is mechanized with a passive hydraulic feedback system and the accelerator pedal is spring mounted (using stock throttle body components). The steering wheel hub contains eight instrumented control buttons that can be assigned any function necessary in the context of a simulation study. A production multifunction turn signal stalk is also instrumented.

The cockpit contains fully functional electric seats with belts, windows, door locks, entertainment center, interior lighting and air flow ducts. Six speakers provide ambient auditory cues for engine and road-tire noise as well as in-vehicle entertainment sound and auditory warnings. Displays include a 10.5" full color flat panel liquid crystal head-down display (HDD) and a 5x10° CRT-based full color head-up display (HUD). The HUD uses custom designed optical components, including a lens assembly, an aspheric mirror and a wedged windshield, to create a large, reconfigurable, color virtual image approximately 2.5 meters from the driver's eye.

Specialized data acquisition capabilities include proximity sensors on the brake and accelerator pedals that use infrared sensors to detect impending contact by the driver's foot. The proximity information obtained supplements the data obtained from the position encoders and enhances our knowledge of driver actions during time critical periods in the near-collision phases of simulated scenarios.

## **2.2 Computational Architecture**

The computing resources used in the system include four Silicon Graphics (SGI) workstations and three IBM-PC compatibles. All of these machines are connected via standard 10 Mbit/second ethernet; the machines responsible for scene generation are additionally connected by a fast shared memory.

The scenario simulation is hosted on an SGI Indy™. As described further below, it communicates with the scene generation software running on an SGI Onyx™. Scene generation is handled by an SGI eight-processor Onyx with one RealityEngine2™ pipe, an SGI four-processor SkyWriter™ with two RealityEngine1™ pipes, and an SGI Crimson™ with VGX™ graphics and a VideoSplitter™ option that generates four separate VGA outputs from the frame buffer. These machines are connected by a VMIC 5550 reflective memory for fast distribution of viewing parameters. This memory has a maximum throughput of 20 Mbytes/second and a minimum access time of 200 nanoseconds.

We use the SkyWriter and Crimson for graphics exclusively. The Onyx, however, hosts the cab data acquisition software and hardware, the vehicle dynamics model, the sensor models, threat assessment, and vehicle terrain-following in addition to providing two channels of graphics output.

### 2.2.1 Image Generation

The system uses a total of nine visual displays. Eight of these are visible to the driver; the ninth is used to run and monitor the scenario simulation and the status of the software components.

Of the driver-visible displays, six are out-the-window scene renderings, and the remaining two are the head-up and head-down displays (Figure 3). Both the HUD and HDD use software written in Tk [2] to provide a rapidly-reconfigurable image that runs in any X11 environment [3]. The HDD is built on the SGI Onyx and displayed remotely on an IBM PC-compatible running X with VGA resolution. The HUD is built and displayed on the SGI Crimson and is one of four windows output in VGA format through the Crimson's VideoSplitter [4]

The six out-the-window scenes are generated on three separate SGI machines. The mapping of machines to displays is shown below.

<u>CHANNEL</u>	<u>DISPLAY</u>
front left channel	SkyWriter pipe 0
front center channel	Onyx pipe 0
front right channel	SkyWriter pipe 1
left rear mirror	Crimson VideoSplitter channel 2
center rear mirror	Crimson VideoSplitter channel 3
right rear mirror	Crimson VideoSplitter channel 4

Each of the forward-looking channels is 1280 pixels wide and 862 pixels high; this size was chosen so that the entire screen area could be used without altering the aspect ratio of the computed images.

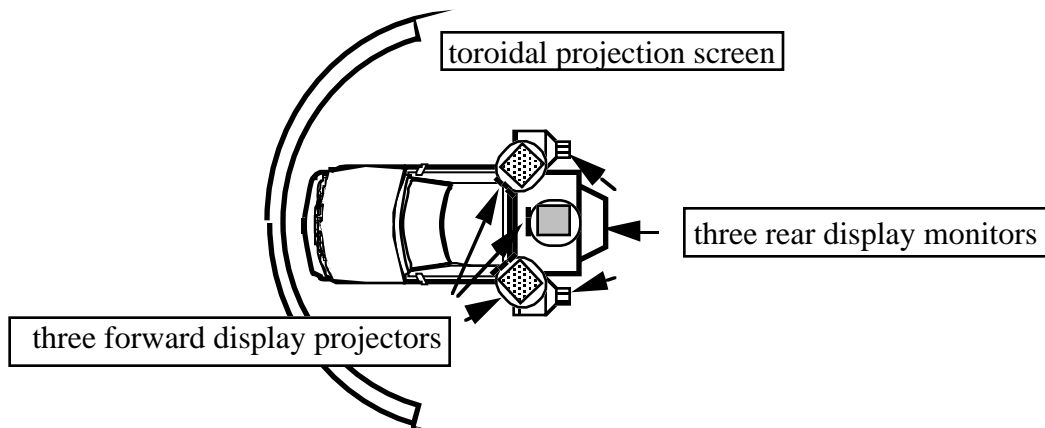
The three forward views are displayed on a 160° toroidal screen, presenting an unbroken panoramic view. The displays are projected from Electrohome Model ECP 4500 high-resolution projectors ceiling-mounted above and behind the cab. The projectors are carefully adjusted to minimize edge effects. Because of the geometry of the projector positions and the toroidal screen, we have determined that it is not possible to compute images in real time that eliminate distortion across the entire screen. However, we do achieve C2 edge continuity.

Each of the rear-looking channels is of VGA resolution. The left and right rear side mirror displays are presented on 17- and 21-inch Mitsubishi color monitors, respectively; these are mounted on stands on either side of the cab behind the position of the front doors. The center rear mirror image is displayed on a Trooper back-projected screen. The rear view displays present imagery of approximately 1.5-2 arc-minutes of resolution to the driver's eyes.

All six out-the-window views are rendered by EasyScene™ from Coryphaeus Software. EasyScene is an SGI Performer™-based rendering system that has been enhanced to work directly with ScenarioBuilder™ from Monterey Technologies for controlling the visual simulation in real time.

EasyScene communicates with ScenarioBuilder via ethernet UDP packets. A frame rate of 30Hz must be maintained for the two applications to work correctly together. Because our current configuration can support this rate only on the Onyx, ScenarioBuilder communicates only with EasyScene on that machine.

This “master” EasyScene performs necessary terrain intersections and channel parameter calculations, then places information in shared memory. EasyScene applications running on the SkyWriter and Crimson read this information at the beginning of their frame cycles, and each then produces images at the highest rate possible. In this way, these slower machines work with the most up-to-date data, eliminate the need for costly intersection testing locally, and are able to run at the frame rate they can best manage. In a future implementation, we hope to generate all six out-the-window views at a fixed 30Hz using a 3-pipe Onyx with the recently-announced InfiniteReality™ graphics.



**Figure 3 Cab and Display System**

### 2.2.2 Sensor Models and Threat Assessment

Each frame, the center forward scene generator passes object position and orientation data through shared memory to a sensor simulation developed by Systems Technology, Inc. This simulation is an optimized multi-beam radar model developed expressly for this application. Every 100 milliseconds, the sensor simulation reads the contents of shared memory and uses it to construct radar cross sections for all objects, both static and dynamic. Dynamic objects include vehicles and pedestrians; static objects include overpass supports and other environmental features.

The output of the sensor simulation is an indication, for each sensor, of whether any object was detected. No identification of the object is provided, and the results are randomly perturbed to produce occasional false positives in the output.

At the conclusion of the sensor simulation’s duty cycle, it signals the threat assessment module, developed by Delco Electronics’ Automotive Electronics Development unit, to interpret the new sensor results. This module uses advanced multi-tracking algorithms to assess the severity of threat from objects detected by the sensors. It attempts to track objects over time, and determines whether they represent real objects or anomalies. If the threat appears real, its trajectory is compared with that of the own vehicle to determine the severity of the threat. The threat assessment software used is, in fact, a straightforward port of the code actually used in test vehicles.

The output of the threat assessment module is an indication, for each sensor, of the presence or absence of a threat, and a categorization of the threat as “caution,” “tailgate,” or “severe.” These outputs are used by the warning systems to determine the characteristics of warnings produced.

### 2.2.3 Scenario Simulation

Simulation scenarios are developed off-line using ScenarioBuilder, which is designed expressly for use with EasyScene. ScenarioBuilder preprocesses the scene database to derive roadway and traffic signal information. Spline control points can then be used to define pathways (lanes, lane changes, and turns)

on the database roadways. Simulated vehicles can be placed along spline paths and provided with acceleration and speed information, as well as information that determines when the vehicle begins movement and how its subsequent movement should be affected by other vehicles, including the own vehicle. For example, a vehicles might accelerate from rest when the own vehicle approaches from behind it, then maintain a fixed distance between itself and the own vehicle.

Such scenarios can be designed in advance and “played out” within ScenarioBuilder using a simulated own vehicle. When the scenarios have been sufficiently validated with this method, ScenarioBuilder can then initiate contact with EasyScene, and through it with the own vehicle’s dynamics software.

### 2.3 Visual Database

Subjects will drive over a single closed-circuit roadway database that includes sections of divided highway, undivided rural highway, suburban roadway, and urban roadway. The database is completely textured, with many environmental objects to add realism and visual complexity.

The divided highway section is four miles long with an estimated driving time of 4.5 minutes. The roadway has three lanes in each direction. There are three simple on/off exits with overpasses. The rural roadway consists of two sections of road, each 1.25 miles long, with an estimated total driving time of 3 minutes. One section is four-lane with no center divider; the other is two-lane with no center divider. The four-lane section contains T-intersections and 2- and 4-way stops. The two lane section contains one T-intersection and 2 stop intersections. The suburban roadway is 1.5 miles long with an estimated 2.5 minutes of driving time. It has two lanes in each direction divided by a center median. The main roadway encounters T-intersections, two- and four-way stop signs, and four-way stop lights. The urban section is .75 miles long with an estimated two minutes of driving time. There are two lanes in each direction with a center median. This section contains T-intersections, four-way stop signs, and four-way stop lights.

### 3. Warning Systems

The car simulator is equipped with three types of warning systems: visual, audio, and haptic. They can be used independently or in any combination. This section describes the available warning systems.

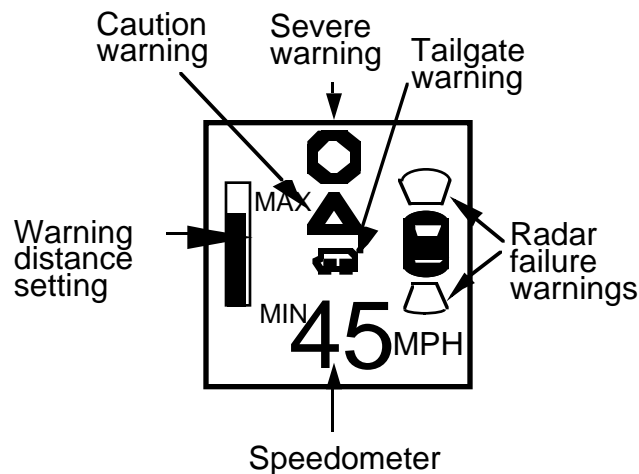


Figure 4 Sample HUD/HDD Display

#### 3.1 Visual Warnings

The visual warnings are displayed on the HDD or HUD. The HDD is a 640 by 480 VGA resolution color flat-panel LCD display, measuring 10.5 inches diagonally. It is mounted in the dashboard in front of the driver, between the windshield and steering wheel. The HUD is driven by a multi-scan color monitor; its optics create a virtual image collimated 2.5 meters in front of the driver’s eyes. This lets the

driver simultaneously watch both the HUD display and the external scene without taking his eyes off the road. The HUD has a maximum field of view of 5° by 10° with a look down angle of 5-6°. The head-up and head-down displays are driven independently. The images are generated as X windows by the SGI Onyx and Crimson. Figure 4 shows a sample HUD/HDD design.

### 3.2 Haptic Warnings

Haptic warning devices provide a force feedback mechanism, using the driver's sense of touch to communicate warnings. The primary haptic warning in this simulator is a "seat stimulator" installed in the driver's chair, underneath the driver's left leg. When activated, the stimulator buzzes, providing a vibration similar to that generated by wireless pocket pagers. Three different levels of vibration can be generated. In the future, we may also install a pushback mechanism on the accelerator that can briefly push the pedal back against the driver's foot, warning the driver to slow down.

### 3.3 Audio Warnings

The audio warnings and sound effects are provided by nine speakers mounted in the cab: three in the dashboard, two in the doors, two in the rear cab, one under the driver's seat and one in the engine compartment. Any six of these may be activated simultaneously. Normally, the woofers in the engine compartment and under the driver's seat provide engine and vibration sound effects, respectively. Four of the remaining speakers provide the entertainment system audio and audio warnings controlled by the simulation software.

Sound effects and audio warnings are generated by playing looped 22 kHz sampled sounds. Because older drivers often have difficulty hearing high-frequency sounds, we avoid including high-frequency components in our warning sounds. Two PC-compatibles, containing two and four sound boards respectively, act as sound servers. These servers accept sound-playing commands from the Onyx via two serial lines running at 38.4 Kbits/second. The sound boards are Crystal River Engineering Alphasatrons™. In addition to generating sounds, these boards can also mix in the output from the cab entertainment system, under the sound server's control. Thus, the warning sounds can be played simultaneously with the other audio on the same four speakers, or the sound server can stop audio from the entertainment system when a warning sound is generated. Generating and mixing the sounds on the PCs is much easier than doing so on the SGIs, and it is also a cheaper solution than tying up expensive workstations for this task. The six outputs from the sound boards are amplified by three amps before being routed into the speakers inside the cab. Figure 5 shows the locations of the speakers inside the cab.

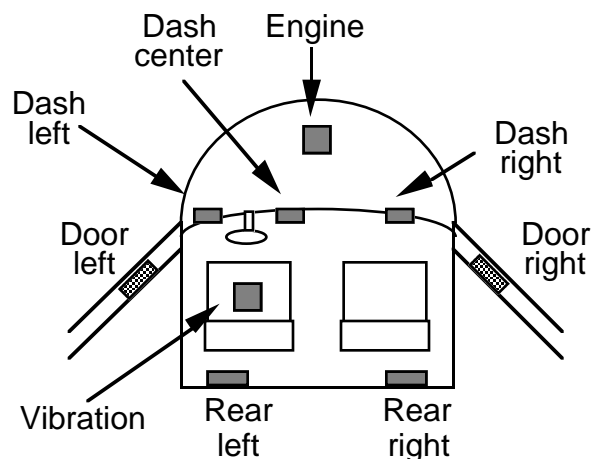
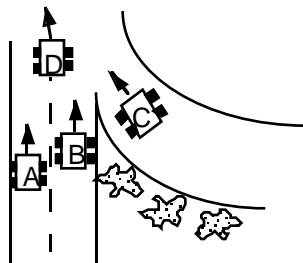


Figure 5 Location of Speakers Inside the Cab

## 4. Scenario Example

Figure 6 shows a typical scenario event. The own vehicle C is merging into highway traffic. A and B are moving straight in their lanes, while D is changing from the right to the left lane.



**Figure 6 "Merge" Scenario**

C is under the subject driver's control, while A, B, and D are controlled by ScenarioBuilder. Their paths, speeds, and interactions have been previously defined in ScenarioBuilder.

At the point in time shown in the figure, the subject sees a visual environment in which he appears to be moving down an entrance ramp, preparing to enter the highway. On the center forward screen, he sees D changing lanes; on the left forward screen, his peripheral vision picks up A and B behind D.

The subject prepares to enter by pushing down on the cab accelerator while maintaining a slight rightward heading using the steering wheel. The accelerator and steering encoders are read 60 times per second by the vehicle dynamics model. The model computes the own vehicle position, heading, and speed, and places these results in shared memory. It also sends this information via ethernet to ScenarioBuilder.

Each frame, EasyScene receives new vectors for all vehicles (except the own vehicle) from ScenarioBuilder. It determines the current viewpoint from the own vehicle data, and computes a new scene for the center forward channel. It places the vehicle information and viewing parameters in the reflective memory, where the "slave" EasyScenes on the Crimson and SkyWriter compute new side-forward and rear scenes

30 times per second, ScenarioBuilder receives new own vehicle data and incorporates them into its knowledge of the ongoing scenario. It computes new position and velocity vectors for A, B, and D and sends them to EasyScene. At the same time, the sensor model picks up the new own vehicle data and determines, for each sensor, whether any other vehicle or stationary object can be detected. Ten times per second, it places these data into shared memory and triggers the threat assessment model.

Threat assessment uses the new data to update or initiate tracks for each potential object. If it determines that an object will come close to intersecting with the own vehicle's trajectory, it places into shared memory an indication of the severity of the threat and activates software that maps the threat indication to one or more warning systems. These systems then produce warnings that are manifested by visual, audio, and/or haptic outputs to the subject in the cab.

## 5. Conclusion

Our goal is to have a fully-functional simulator that can reliably run for hours at a time with minimal intervention, and that does not require a trained computer scientist to control and program. We believe we have largely met this goal by compromising elements of the system that do not directly affect the goals of our crash avoidance study. Accordingly, we have used a fixed-base simulator, a fairly simple and flat visual database, and a hardware suite chosen for cost-effectiveness and ease of use rather than absolute performance. The simulator described here is a "work in progress" that will continue to improve over the course of subject testing. In particular, better synchronization among the six out-the-window scenes will be important in reducing simulator sickness and in reducing perceived lag between events and warnings. It is also likely that lessons learned early in subject testing will demand substantial changes in the warning system -- indeed, that is an important motivation for constructing the simulator. Finally, as field tests provide better sensor information and improved threat assessment algorithms, these must be incorporated into the existing system. It is critical that simulator users understand that the performance of components



and the relationship between subsystems must be considered in light of the human at the center of the system. Take care in setting up the simulator, make changes with caution, and watch your test drivers closely!

## References

---

- 0 Jon Kuhl, Douglas Evans, Yiannis Papelis, Richard Romano, and Ginger Watson. The Iowa Driving Simulator: An Immersive Research Environment. *IEEE Computer* 28, 7 (July 1995), 35-41
- 1 Rod Deyo, John A. Biggs, and Pete Doenges. Getting Graphics in Gear: Graphics and Dynamics in Driving Simulation. *Proceedings of SIGGRAPH '88* (Atlanta, GA) 1-5 August 1988. In *Computer Graphics* 22, 4 (August 1988), 317-326.
- 2 John K. Ousterhout Tcl and the Tk Toolkit. Addison-Wesley 1994.
- 3 Adrian Nye. *Xlib Programming Manual*, Volumes One and Two, O'Reilly & Associates, Inc. 1990.
- 4 Silicon Graphics, *VideoSplitter Programmer's Guide*, version 1.0. Document Number 007-5352-010, February, 1991.