Advanced Rendering Cluster for Highway Experimental Research (ARCHER)

Jason R. Williams
AAI Services
Jason.williams@fhwa.dot.gov
(202)493-3392

Tsai-Chia (Peter) Chou
AAI Services
peter.chou@fhwa.dot.gov
(202)493-3087

Barry L. Wallick
AAI Services
wallicbl@aaicorp.com
(410) 628-3837

M. Joseph Moyer
Federal Highway Administration (FHWA)
Joe.Moyer@fhwa.dot.gov
(202) 493-3370

Turner Fairbank Highway Research Center
Federal Highway Administration
6300 Georgetown Pike
McLean, VA 22101

Abstract

The Federal Highway Administration (FHWA) Highway Driving Simulator (HDS) has been undergoing a major upgrade to a PC-based cluster rendering system. This new system is dubbed “Advanced Rendering Cluster for Highway Experimental Research,” or ARCHER. ARCHER is written in C++ and Java™, and utilizes open-source libraries such as OpenGL™, sgl (scene graph library), and cg™ (dynamic shading). Linux is the primary platform and RealTime™ technology products, Generic Vehicle Dynamics System™ (GVDS) and the MotionTilt™ library, are used for vehicle dynamics and motion base controls.

The challenge was to create a flexible and upgradeable system with multi-channeled projector support, while still maintaining a high frame rate for scenes containing highly complex roadway and roadside geometry. Night and day scenarios, realistic vegetation, retro-reflective signs and pavement markings, and the capability to create new non-standard traffic control devices, signs, and pavement markings are all critical capabilities of the new system. The new scenario creation methods used in ARCHER are based on layered databases, which allow the mixing of real and artificial sources of data. ARCHER has the capability to use geographic databases, such as the United States Geological Survey (USGS) digital

elevation and image databases, in order to quickly create realistic scenes based on actual roadways under study. The roadways themselves can be ingested from either GPS traces from a driven vehicle, or for high-resolution geometry, from the FHWA's Digital Highway Measurement (DHM) van. The DHM is a suite of instruments on a single vehicle platform designed to provide accurate and efficient measurements of roadway geometry, roadside hardware and pavement conditions. Such accurate field measurements significantly improve the ability of researchers to compare driver behavior derived from real-world roadway data with data collected in a simulator. Recently the DHM van was used to collect roadway and roadside data to create a model of a Pennsylvania road for a joint Pennsylvania Department of Transportation (PennDOT) FHWA Highway Driving Simulator study.

This paper describes how the ARCHER system was conceived and implemented to support FHWA behavioral research on highway safety and operations.

Introduction


Fig. 1 Highway Driving Simulator

The Federal Highway Administration's (FHWA's) Highway Driving Simulator (HDS) at the Turner Fairbank Highway Research Center (TFHRC) has been undergoing an upgrade for the past two years. The first stage of the upgrade was completed last spring in 2005, in time to run a novel experiment in cooperation between the Pennsylvania State Department of Transportation (PennDOT) and FHWA. The upgrade was essentially a full re-design of the hardware and software systems of the simulator to better accomplish the goals and requirements of the Human Centered Systems (HCS) team at TFHRC. On the hardware side, the original Onyx 2 system was replaced with a cluster of PCs with individual PCs functioning in the following roles: Image Generators, Vehicle Dynamics, Scenario and Scene Control, Operations Control (OpsCon) GUI, Motion and Car Cab I/O Controller, Data Collection, and Audio. A Projector upgrade to a 240 degree wide field-of-view (FOV) using Barco 909 projectors was also completed during this time (see **Fig. 1**). The newly designed system was dubbed "Advanced Rendering Cluster for Highway Experimental Research," or ARCHER. The PA Route 851 (PA851) cooperative study between PennDOT and FHWA was used as part of the functional requirements and as the final test of the upgraded ARCHER system.

Design Requirements

The research conducted by the HCS team in the HDS can vary widely (**1,2**), from pavement marking research to innovative traffic control devices or even to visually impaired pedestrian cross-walk research. In each of these experiment types, a driver in the simulator is given visual, auditory, and tactile (or motion) stimuli to elicit driving behaviour, which must closely match how that driver would drive in the real world. The new simulator system needs to be flexible and allow for inexpensive hardware upgrades as needed. A scenario construction capability that is extensible is a primary focus, as new and unique scene content will be needed for each new experiment. As a Highway Driving Simulator, ARCHER needs to render scenes at a real time rate of at least 30-60 fps for each view (front, sides, rear) around the car cab. The system needs to collect data at this same rate on the vehicle position, orientation, lane keeping, vehicle controls, as well as on highway signal device states. The system has to generate sound effects consistent with the roadway environment, including but not limited to engine, road/tire, wind, tire squeals, and other traffic sounds. The geometry for a roadway in the scene may come from real geo-specific sources or from geo-typical sources as each experiment may require. The PA851 experiment requires the replication of six miles of real roadway that was scanned using the FHWA's Digital Highway Measurement van (DHM) equipped with a range finding laser system and returned data at the sub-inch level (see

Fig. 4) (**3**). The system has to generate pavement markings and road signs, which have retro-reflective properties to be displayed during nighttime scenarios. In addition, for night time studies a realistic headlight pattern needs to be generated as well as the associated effects for modern traffic signage at night. Sources for terrain surface topology may come from real data using geographic information system (GIS) databases or direct data sources such as U.S. Geological Survey (USGS) Digital Elevation Map (DEM) sources. Imagery for this geography can be ortho-rectified satellite or aerial photography. Realistic buildings and foliage need to be rendered in scenes such that appropriate view distances are maintained in the scenarios. A graphical user interface (GUI) is required that supports the scenario controls for the experiment, as well as playback and analysis capability. Scenarios need to be definable in such a way that allows different trials to be run in which the same roadway has different treatments or traffic control devices to be tested.

Top-Level Software Architecture

The software architecture of ARCHER has been constructed to support scalability, extendibility, flexibility, and evolveability. The software is scalable in that ARCHER can run on either one single workstation or on a cluster of workstations of as many nodes as is available (**4**). It is extensible in that ARCHER can add new functions and scene content for each new experiment as it is being defined. It is flexible in that ARCHER can easily be modified, or new components can be implemented, while still reusing most of the existing code-base, to carry out each new experiment. It is evolvable in that ARCHER can directly run on new hardware, particularly, new graphics cards, for better performance without modifying existing code. To be successful in providing these salient characteristics, ARCHER has been implemented in the object-oriented programming languages, C++ and JAVA™, and uses OpenGL™, an industry standard graphics application programming interface, as well as utilizes open-source third-party libraries as much as possible.

From a programming point of view, the software architecture of ARCHER consists of four components: GUI, Viewer, Rendering, and Devices. The GUI component provides human-computer interface for an operator to conduct an experiment. The Viewer component generates and distributes viewing parameters as well as rendering commands to the associated components. The Rendering component maintains the scene contents and renders perspective views in real-time. The Devices component conducts vehicle dynamics as well as drives the motion-platform, car-cab, desktop steering wheel, and audio equipment. The framework of these four components is described in the following sections and is shown in **Fig. 2**.

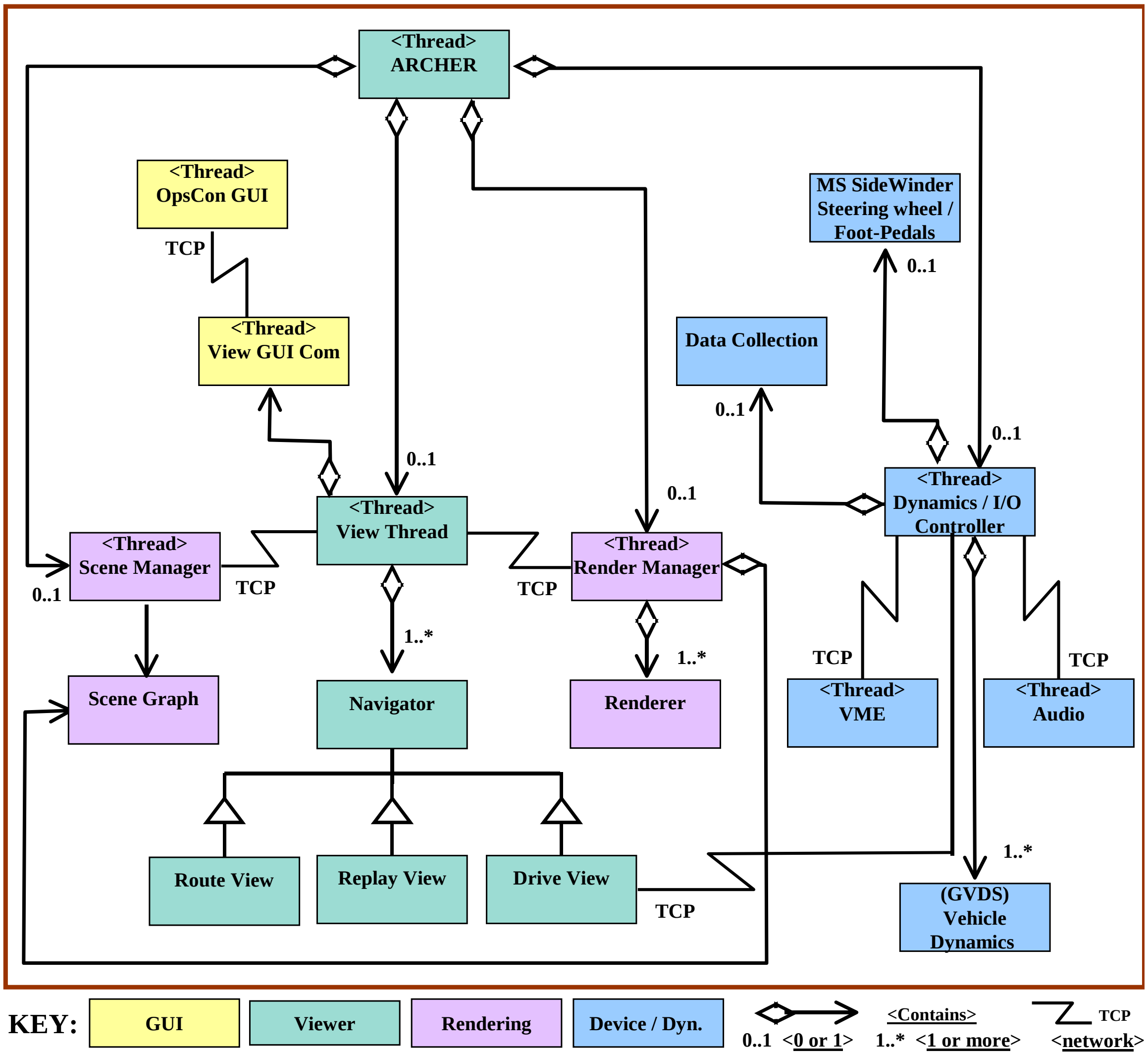

Fig. 2 ARCHER Top Level Architecture

Operators Control (OpsCon) GUI

ARCHER GUI provides a panel for the operator to start and stop a driving session, to select the roadway treatment of a driving session, to record the rating from the research participant, and to replay a previously driven scenario. The GUI also allows the operator to interactively change certain rendering features, such as lighting, position, and hiding scene object type, such as embankments, trees, etc. Scenario status is displayed for both actively driven sessions as well as replayed scenarios. VCR like controls allow for reviewing the replayed scenarios.

View Control / Rendering Subsystem

The ARCHER system can be set-up to run with one or many view windows. The view windows can be configured to display Front, Side or Rear views from the driver's perspective, or can be used as plan views, or as tethered God's Eye perspectives of the roadway scene. A view can be assigned to a single computer or many views can be run on the same computer. The eye-points for each view are controlled from a single View Controller, which gets positional updates from the Vehicle dynamics system. The View Controller uses network calls to synchronize each view on a frame-by-frame basis for rendering frame swaps.

Dynamics - I/O Subsystem

The dynamics module gets driving inputs (acceleration, braking, and steering) from a VME interface to the simulator's car-cab, or from a Microsoft Side Winder™ steering wheel, which can be substituted for testing or desktop simulation. These inputs are fed into the vehicle dynamics library (GVDS™ from RealTime™ Technologies) to update vehicle position, attitude, speed, etc. The vehicle dynamics subsystem sends the updated view parameters to each computer in the cluster through the view control interface. The dynamics module also sends audio parameters to the audio package for generating sound effects. Output values are sent back through the VME interface to drive the motion-platform and the display dials in the car-cab.

Audio Subsystem

The new HDS audio system replaces the pre-existing 1980's analog hardware system with a digital multi-channel hardware using WAV files. A single Audigy™ 2 audio board capable of producing up to 7.1 channels is installed in a standard PC with a network connection. An existing 5.1 multi-channel audio amplifier (100 watts/channel) is used in the simulator for audio playback. The software creates five Sound buffers to simulate engine, tire, wind and tire squeal, and reserves one buffer for future expansion. These sound buffers are modified each simulation frame (typically 60Hz) by software control based on parameters received over the network connection. Each frame, the network passes accelerator pedal depression, engine RPM, tire slippage, and vehicle speed data from the vehicle dynamics module (running in a separate PC) to the audio PC. These real time parameters are used to create interactive audio feedback for the driver. Software functions within the Sound class are employed to modify the frequency and volume of each sound buffer based on inputs from the real time simulation. This provides real-time sound effects, which reflect the real time vehicle parameters.

Scene Management Subsystem

The Scene Management subsystem is responsible for scene and scenario loading, updates, and control. Scenario content is classified into unique classes based on rendering and updating similarities. A separate loading and updating C++ class called a Scene Object Creator is created for each type defined. These creators are organized in a C++ class called the Scene Object Factory, from which new scene objects of the specified type are generated as they are read in from the scenario file. **Table 1** illustrates the current types of Scene Objects, which can be defined in an ARCHER scenario:

Table 1 Scenario and Scene Object Types

Road Objects	Terrain	Vegetation	Models
Road Surface - GPS	USGS DEM	Trees	Houses
Road Surface - CAD	Dig.OrthoPhotos	Shrubs	Vehicles
Rd Surf. DHM	Aerial Photos	Bushes	Pedestrians
Pavement marks (PM)	Satellite Photos	Grass	Animals
Retro-Reflective PM	Height Field	SpeedTree™	Misc.
Horizontal signage			
In Pavement LEDs			
Rumble Strips			
Speed Bumps	**Embankments**	**Lights**	**Events**
Guard rails	Cross-sections	Sun Light	Distance
PMD - Reflectors	Custom	Head Light	Lane Tracking
Side Roads			Timers (lights)
Signal Lights			Ped.Xing Event
			Vehicle Events

Road Objects

Road Objects are all relative to the roadway for positioning. The road geometry is called the Road Surface. The road surface can be defined by several techniques including using GPS or CAD files, or, as in the PA851 experiment, using the FHWA's DHM van (**3**)(see **Fig. 3**). The DHM uses multiple instruments to scan the road surface. The resulting data can be converted into a road surface data file (see **Fig. 4)**. Pavement markings are defined with the road geometry data file and the properties (color, wear, reflectance) can be controlled separately to generate experimental trials. Retro-Reflective Pavement Markers are also modeled and placed on the road surface. Placement of Guardrails is supported, and these include reflectors for nighttime scenarios. Intersection side roads and Signal heads can be placed in the scenario. Rumble strips, in pavement LEDs and horizontal signage have been used in ARCHER scenarios and are modeled.

Fig. 3 DHM Vehicle

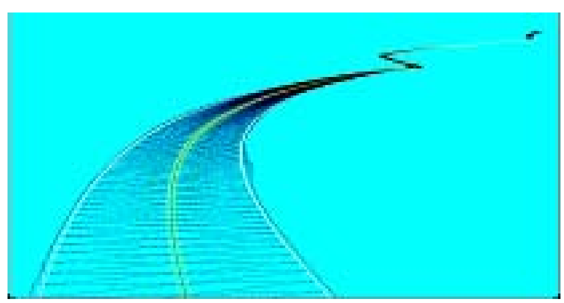

Fig. 4 DHM Derived Road

Terrain

Sources of terrain data include digital elevation maps and imagery supplied by the USGS. Alternatively terrain can be constructed from height maps as needed for non-geo-specific scenarios. The PA851 experiment used tiles from the 10m USGS DEM data and 1 m aerial imagery for placement and coloration.

Embankments

Embankments are especially important to fill in the gaps left by the relatively low resolution of geo-specific data sources such as USGS 10m DEM data. The embankments are representative of cross-sections perpendicular to the roadway at semi-periodic intervals. The embankments for the PA851 experiment were manually created based on video of the PA 851 roadway.

Vegetation

Many scenarios require dense vegetation to correctly block view distance down a roadway. The ARCHER system has taken advantage of the SpeedTree™ library of real-time tree and vegetation rendering. This library allows the rendering of highly realistic trees and vegetation using aggressive level of detail modeling and controls.

Lights

Both daytime and realistic nighttime shading is supported in ARCHER. The nighttime headlight algorithms were created using existing headlight measurements as the basis, and using the TARVIP lighting model in an offline manner to calibrate the resulting scene (**5**). The headlight algorithms use separate lighting models and controls to render the highly reflective and retro-reflective materials, and separate algorithms for the diffuse materials. Graphics card hardware acceleration shader libraries known as cg™ are used to model the retro-reflective materials in real-time. **Fig. 5** and **6** show example night scenes in which retro-reflective signs and pavement markers play a significant role.

Fig. 5 Headlight Scene 1

Fig. 6 Headlight scene 2

3D Models

Pre-existing or specially crafted 3D models can be used to represent many of the scene objects in ARCHER. Loaders exist to natively read Creator OpenFlight (FLT) models as well as 3D studio files (3ds). 3D Models are generally used for buildings, vehicles, pedestrians, and other misc. non-terrain or road surface objects.

Events

Various kinds of events can be used in scenario definition and in data collection. Events can be triggered by distance checks, lane-keeping values or by timers (periodic or absolute). Events can spawn pre-define behavior of scene objects or can be used when to trigger data collection.

Data Collection

Data collection can either be frame by frame with a predetermined set of variables, which is configurable for each experiment, or by data collection events which can collect predefined variables based on a triggering events in the scenario such as distance to a signal head.

Summary

Fig. 7 Real Scene

Fig. 8 Simulation

The ARCHER system has improved the HDS vastly over the predecessor HYSIM system. Experiments are now highly realistic, tightly calibrated, and generated more quickly using much more sophisticated tools than before. In **Figs. 7** and **8,** the simulated roadway closely matches the real road. The future of the ARCHER system should see additional GUI development for even more rapid interactive scenario generation. Other future work will include more night-time headlight modelling by collaborating with the TARVIP model creators to create better hardware assisted headlight shaders. Of course, the ARCHER system will also evolve as needed to better support the FHWA highway research as each new experiment generates its own unique and challenging requirements.

Acknowledgements

The authors would like to thank the following people and organizations that contributed to the development of the ARCHER system and the upgrade of the HDS. FHWA HCS team members Tom Granda, Joe Moyer, and Greg Davis obtained the HDS funding for an upgrade. The support of Michael Trentacoste the Director of the Office of Safety R&D at TFHRC is much appreciated. The authors would also like to thank all the hard work by the past and present AAI staff at the HDS including Duoduo Liao, Mike Baumgartner, Eric Wang, Ryan Cartwright, each played a major role in creating the next generation driving simulator. The authors would also like to acknowledge the SAIC part of the HCS team in helping define and create the ARCHER system. The SAIC scientists, including John Molino, Vaughan Inman, Bryan Katz, and Sheryl Miller, generated the requirements for the new HDS. Thanks also to SAIC team members Lindsay Clark and Jason Kennedy, and to SAIC Project Manager Steve Fleger. A special thanks goes to Dana Duke (SAIC), who helped us with many difficulties relating to the old analog hardware.

References

1. Molino, J., Katz, B., Duke, D., Opiela, K., Andersen, C., and Moyer, M. “Field Validation Study for the Relative Luminance of Pavement Markings and RRPMs at Night,” Proceedings of TRB 83rd Annual Meeting, Washington, D.C., January 11-15, 2004.

2. Molino, J., Katz, B., Duke, D., Opiela, K., Andersen, C., and Moyer, M. “Validate First; Simulate Later: A New Approach Used at the FHWA Highway Driving Simulator,” Proceedings of Driver Simulation Conference, North America, Orando, FL, Nov 30- Dec 2, 2005 (under review).

3. Gagarin, N and Mekemson, J.R. "Executive Summary: Repeatability and Accuracy of Centerline Geometry Measured by the Digital Highway Measurement Vehicle," Task Order Report, Turner-Fairbank Highway Research Center, McLean, VA, April 2005 (under review).

4. Chen, Y., Chen, H., Clark, D.W., Liu, Z., Wallace, G., Li, K., “Software Environments For Cluster-based Display Systems” Department of Computer Science, Princeton University, Princeton, NJ.

5. FHWA-HRT-04-141, Enhanced Night Visibility Series: Volume X, Phase II--Visual Performance Simulation Software for Objects and Traffic Control Devices (under review).