



National University Rail Center - NURail
US DOT OST-R Tier 1 University Transportation Center

NURail Project ID: NURail2012-UIC-R04

Immersive Visualization of Rail Simulation Data

By

Jason Leigh
Professor

Department of Computer Science
University of Illinois at Chicago
spiff@uic.edu

Vaibhav Govilkar
Master student

Department of Computer Science
University of Illinois at Chicago
vaibhavgovilkar88@gmail.com

Arthur Nishimoto

Graduate research assistant
Department of Computer Science
University of Illinois at Chicago
anishi2@uic.edu

Sangyoon Lee

Graduate research assistant
Department of Computer Science
University of Illinois at Chicago
slee14@uic.edu

Lance Long

Senior Research Programmer
Department of Computer Science
University of Illinois at Chicago
llong4@uic.edu

Luc Renambot

Research Associate Professor
Department of Computer Science
University of Illinois at Chicago
renambot@uic.edu

Grant Number: DTRT12-G-UTC18

Final Report to the Office of the Assistant Secretary for Research and Technology

Project: Immersive Visualization of Rail Simulation Data

Grant period: 5/1/12 – 1/31/14

Submitted: 8/31/15

PI: Jason Leigh

Director, Electronic Visualization Laboratory

Professor, Department of Computer Science

University of Illinois at Chicago

Research project (100% of budget)

Participants:

Students supported:

Vaibhav Govilkar – MS student

Arthur Nishimoto – MS student

Sangyoon Lee – Ph.D. student

Staff members supported:

Lance Long

Luc Renambot

Accomplishments

The prime objective of this project was to create scientific, immersive visualizations of a Rail-simulation. This project is a part of a larger initiative that consists of three distinct parts. The first step consists of performing a finite element analysis to determine modal stiffness values and mode shapes of a particular railway track and substructure geometry. The second step involves running a simulation using this information to analyze the effects of a locomotive passive over this substructure. The third step involves creating an 'Immersive Visualization of the Rail Simulation Data' using the CAVE2 [CAVE2], the central scope of this project.

Introduction

The first two sections of the RailSim initiative consist of work done by students of the Mechanical and Civil Engineering Departments. The visualization itself was handled by the Electronic Visualization Laboratory with support from Mechanical Engineering. A large part of the project involved converting large data-sets into scientific visualizations that an engineer may observe, assess, analyze in a meaningful way with the goal of drawing conclusions aided by a superior understanding of how the objects in the simulation interact.

The simulation itself is run on a geometry constructed in the finite element analysis tool ANSYS. It consists of the following major bodies:

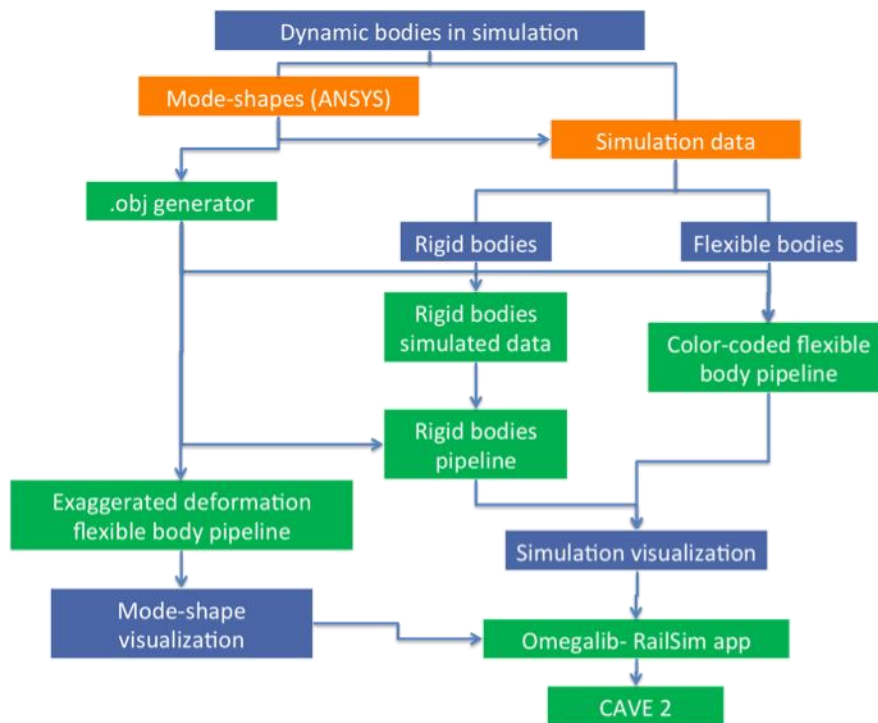
- Rigid rails
- Deformable rail
- Rigid sleepers
- Ballast (Upper gravel)
- Sub-ballast (Middle soil)
- Sub-grade (Lower soil)

- Fasteners (Spring elements)
- Suspended wheel-set (representing a loaded locomotive)

The data from ANSYS is used as an input to a mechanical simulation software SAMs to run multibody dynamics simulations of a train or suspended wheelset over the track geometry. These simulations in turn produced new simulation data sets.

The data sets from ANSYS and the simulation were used to create visualization. They are put through several stages of pre-processing before being passed into the CAVE2 application.

Work Flow



Where:

Project work

Green: Work done as a part of the visualization project.

External work

Orange: Tasks completed externally primarily by the mechanical engineers to provide the input data for the visualization.

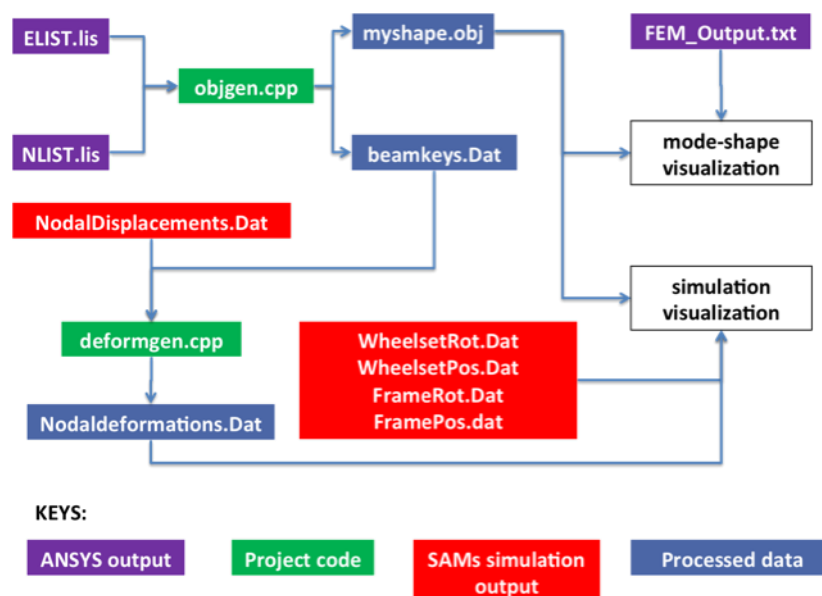
Description of the work flow

- Modal stiffness values and shapes are generated in ANSYS and its data is passed on to SAMs to run simulations.
- The ANSYS data is passed into the .obj generator to create 3d models in the form of .obj files.
- The .obj files in conjunction with the ANSYS data is passed to the 'Exaggerated deformation flexible body pipeline', which will be used in the Mode-shape visualization.
- The .obj files in conjunction with the SAMs simulation data is passed to the rigid body, and color-coded flexible body pipeline to achieve a visualization of the simulation itself.
- The OmegaLib [OmegaLib] RailSim app for the CAVE2 handles these two visualizations to be displayed on the CAVE2.

Code Base

The code base to achieve the given workflow consists of the following:

Pre-processing code

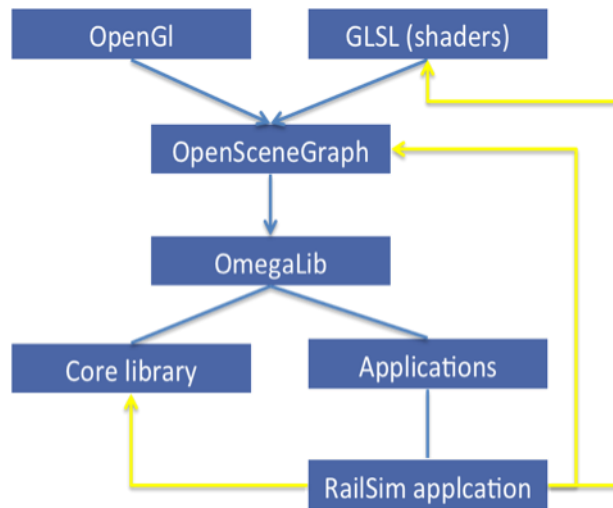


- **objgen.cpp**
This file is used to generate the mode-shape itself. The mode-shape consists the railway track and the substructure below it. It uses two files - a list of nodes and a list of element definitions to construct the .obj file. It also constructs an intermediate file 'beamkeys.Dat'. This files is used to generate deformation data for extra vertices added during the construction of the rail and sleeper cross-sections
- **deformgen.cpp**
This file is used to generate a file called 'Nodaldeformations.Dat' using the output from the SAMs simulation. This file uses 3D vector deformations of every node to generate a file of the net magnitudes. This output is finally passed into the color-coded flexible body pipeline to visualize the simulation. The intermediate data file 'beamkeys.Dat' is used to add deformation values for the newly generated vertices in the .obj files which are a part of the sleeper and beam cross-sections.
- The **mode-shape visualization** uses the .obj file and the 'FEM_Output' or the Finite Element Mode output from ANSYS for its pipeline as inputs.
- The **simulation visualization** uses the .obj files. The suspended wheelset / frame's rotation and position generated by SAM along with the processed file 'Nodaldeformations.Dat' as inputs to its pipeline.

The OmegaLib application

- The RailSim application uses the OmegaLib library along with several OpenSceneGraph [OSG] calls to visualize the mode-shape as well as the simulation.
- The app codebase also includes custom shaders to manipulate attributes of individual vertices within the mode-shape.

The application makes calls from several sources, best described in the following chart.



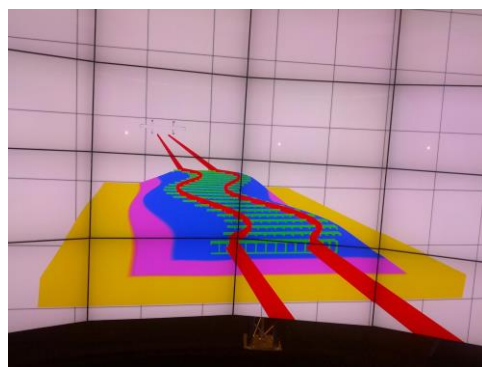
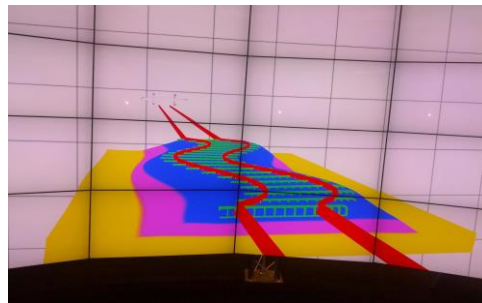
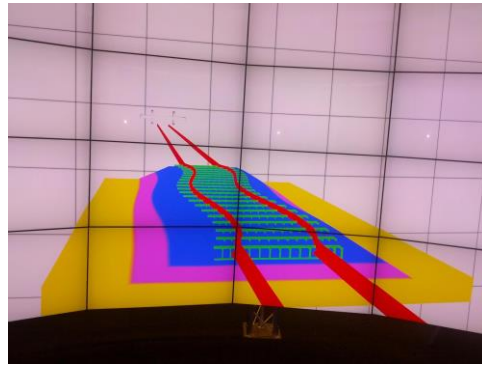
The yellow lines represent the sources from which the application derives its code.

- The OmegaLib library is built upon OpenSceneGraph, which in turn is written in C++ using OpenGL. OpenSceneGraph also allows direct access to shaders through several of its inbuilt function calls.
- The OmegaLib library as it currently exists is divided into the Core Library, which contains all the functionalities, and the applications library which contains applications that use the core library.
- OmegaLib allows the application developer to make direct OpenSceneGraph calls if required.
- The RailSim application (contained in the applications library) uses primarily the core library code. But also uses several OpenSceneGraph functionalities, as it requires the manipulation of geometries.
- The manipulation of geometries requires altering specific vertex attributes.
- Hence the application uses custom shaders that allow vertex manipulation written in GLSL that receive input from the OpenSceneGraph calls.

Mode-shape visualization

The mode-shape visualization uses the 'Exaggerated deformation flexible body pipeline'. After the application loads the .obj file of the mode-shape the pipe-line passes the deformations-per-vertex for the mode-shapes into the vertex shader and displaces them with a multiplicative magnifying factor that increases from 0 to 150 over time and then loops back to 0. This distorts the mode-shape in an exaggerated fashion to describe the current 'mode' selected.

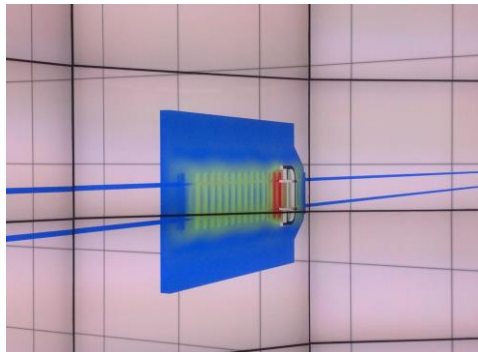
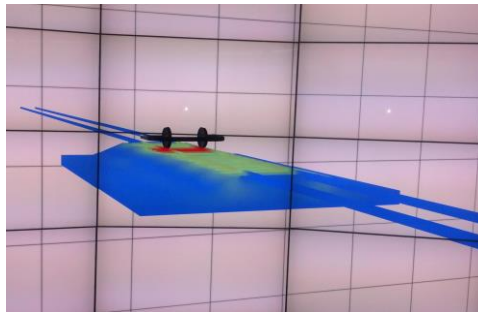
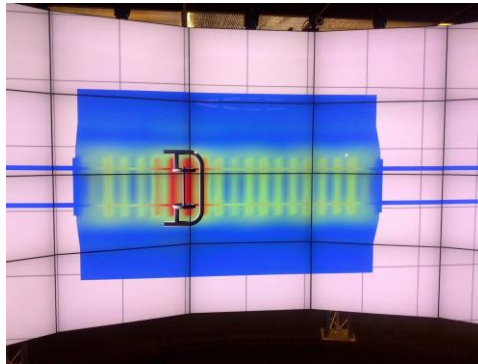
Following are a few images of a mode-shape with varying degrees of exaggeration in its deformation:



Simulation visualization

The simulation visualization uses the 'Color-coded flexible body pipeline'. The application loads the .obj file of the object (same as the mode-shape visualization). It also loads the wheel-set positions and rotations per-time-step and the nodal-deformations of the object (rails and substructure) as the wheels pass over it from 'Nodaldeformation.Dat'. The per-vertex deformations of the object as time progresses are encoded in the form of per-vertex coloring. The colors vary from blue (for static nodes) to red (maximum deformation). This spectrum allows us to see the effects of the loaded wheelset on the rails and the sub-structure as it passes over-head.

Following are a few images of the simulation from various perspectives. As can be seen the region on the rail-substructure in contact and closest to the wheel-set shows maximum deformation signified by the red color.



The application also includes the following features:

- Two modes: Simulation mode, mode-shape mode.
- Fully dynamic camera with free flying as well as smooth transitioning to pre-fixed positions
- Pause, Play, Step-over individual time-steps.
- An interactive menu-system.

The PlayStation move controller is used to control the simulation. Each button can be custom-mapped to a required functionality. Currently the controls are as follows:



The project code has been documented to allow future students to build upon the existing code base as the larger RailSim initiative grows in scope. The existing code can be used for new simulation data as the data-form remains consistent.

Products

Vaibhav Govilkar, Immersive Visualization of Rail Simulation Data, M.S. Project, Department of Computer Science, University of Illinois at Chicago, 2013

References

CAVE2 - Febretti, A., Nishimoto, A., Thigpen, T., Tolandis, J., Long, L., Pirtle, J., Peterka, T., Verlo, A., Brown, M., Plepys, D., Sandin, D., Renambot, L., Johnson, A., Leigh, J., CAVE2: a hybrid reality environment for immersive simulation and information analysis, in the proceedings of SPIE 2013 - The Engineering Reality of Virtual Reality 2013, San Francisco, CA, February 3-7, 2013.

OmegaLib - Febretti, A., Nishimoto, A., Matevitsi, V., Renambot, L., Johnson, A., Leigh, J., OmegaLib: a Multi- View Application Framework for Hybrid Reality Environments, In the proceedings of IEEE Virtual Reality (IEEE VR 2014), Minneapolis, MN, March 29 - April 2, 2014.

OSG - <http://www.openscenegraph.com>