Investigating the Post Processing of LS-DYNA® in a Fully Immersive Workflow Environment

Ed Helwig[1], Facundo Del Pin[2]

[1]*Livermore Software Technology Corporation, Livermore CA*
[2]*Livermore Software Technology Corporation, Livermore CA*

Abstract

The use of virtual reality (VR) in engineering applications has been expanding for the last decade. Immersive technology is quickly becoming a tool for pre and postproduction decision-making and analysis. Virtual reality can assist in reducing the number of physical prototypes, build collaboration between various engineering disciplines, speed up time to manufacturing, and reduce the number of design cycles. We examined the integration of LS-DYNA into a workflow using results from a fluid-structure interaction problem. The expected outcome was to generate life like 2D and 3D simulation models, while maintaining a high degree of engineering data in the analysis output. Additionally, simulation data was placed in a computer aided virtual environment (CAVE) using a passive visualization solution, and eliminating the requirement for an active VR headset. The investigation identified key hardware and software considerations while optimizing the workflow process. Scalability, computation time, component costs and functionality were variables considered during development. It is our firm belief that seamlessly integrated visualization tools and state of the art physics solvers are in the core of future design and manufacturing pipelines.

Introduction

The initial work that we completed for investigating our method to post process LS-DYNA with an immersive environment is just one plausible solution among many possibilities. Our primary drive was to minimize the hardware and software costs, while keeping the methodology focused on existing Open Source and research tools. Established frameworks such as CalVR[1], FreeVR[2] and OpenSceneGraph[3] provided an excellent starting point for our investigation.

Rendering an interactive environment to reflect real world problems provides its own challenges between engineering disciplines and decision makers. There is a fine balance between environments that reflect reality for all users, while trying to maintain the engineering value for analysis. We chose to concentrate on the methodology, but allowed this to influence some of the key focal points of our investigation.

We identified several fluid features from our Computation Fluid Dynamics solver that we wanted to visualize in an immersive environment and LS-PrePost®. Over the past three years, we have designed and constructed a framework that could support our initial goal. We worked with customers, researchers, and companies already established and experienced in Virtual Reality and CAVE environments. The following sections outline the general approach to our methodology, and provides examples that will further our investigation.

Related work

A commercial virtual reality system supplier had interest in creating a tool for High Dynamic Range (HDR) static images and animations from LS-DYNA results. LS-PrePost[4] offered a set of APIs to extract modeling surfaces and results for their workflow. These developments were limited to structural simulations, and their tools did not offer an Open Source framework after the product development. Our investigation is mainly focused on Computation Fluid Dynamics results, but our intent is to eventually include all the possible LS-DYNA solver results. Previous research and experience with the supplier provided valuable insight as to how we could expand on this idea, and enhance our capabilities for future work. We wanted to minimize the costs and reduce the artistic man-hours for rendering and post processing.

We also believed that having different tools for visualizing data in many ways often provides additional insight and feedback beyond the standard conventions for the user.[10] Generally speaking, the scientific data is highly dimensional, and is best represented in 3D. We selected specific CFD analysis functions that could be developed for LS-PrePost 4.0 (*fig1*) and a CAVE system (*fig2*). Visualizing vortex cores, separation/attachment lines, iso-surfaces, cutting planes and Line Integral Convolution (LIC) were implemented for 2D and 3D visualization with success during our investigation.

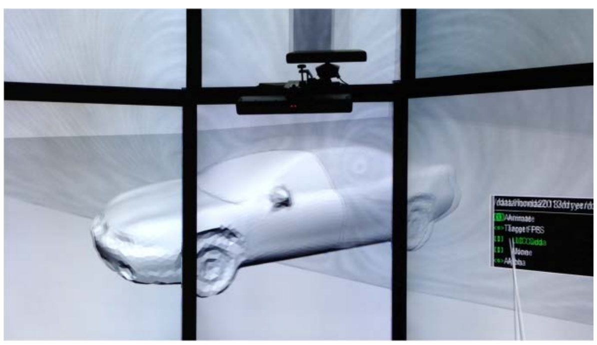

Figure 1

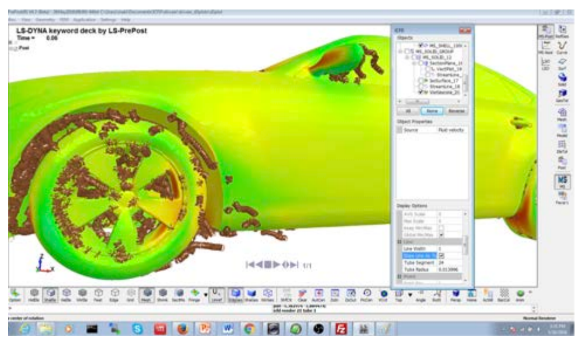

Figure 2

Hardware and software

The hardware specifications for the CAVE system included LG 65-inch 2160p Flat OLED 4K TV, Dell Precision 7000 Tower (192GB DDR3, 2TB SATA HD, 1300W Power Supply), NVIDIA GTX 1080 graphics adapters, HTC Vive, supporting cables, mounts and RealD 3D glasses. Similar specifications can be found in the Next Cave[5] and The Wall systems currently in production at Calit2, UCSD and other locations.

The CalVR framework makes the system flexible for creating a virtual reality environment. It supports many non-standard VR system configurations, and enables the use of consumer grade hardware, displays and input devices. It was developed for the Centos Linux and MacOS platforms, but can work with most major distributions. Multiple user inputs support network collaboration between remote locations, a modular plug-in system separate from the main code base, sound and head tracking support capabilities provide a complete virtual reality experience. CalVR uses OpenSceneGraph as its graphical library.

Rendering hardware for the master and slave nodes consisted of generic Centos Linux workstations, quad core 2.67 MHz processors, 64GB of memory, and Nvidia Quadro 600 (96 cores/1024MB) graphic adapters. The rendering job control was integrated with the Blender Network[8] to support a clustered type environment. Each compute node image file was a self-contained ISO with the Linux OS and all necessary tools to extract and render its assigned scene.

Model and surface extraction from the LS-DYNA output was accomplished with Paraview[6] and rendered with Blender[7]. Both Open Source packages use Python modules that could be executed remotely in batch mode. LS-DYNA results for the CalVR environment were extracted from VTK output, and processed with a CalVR plugins specifically defined to view the CFD functionality in 3D space.

Navigation in the CAVE is accomplished through lower level programming by the CalVR framework to an HTC Vive controller. Position and tracking data are not used, so the point of view is fixed for each observer. The HTC joystick provides a pointer mechanism in the defined 3D space to manipulate objects and movement (fig3). Additionally, the CalVR Board Menu is created with a custom XML configuration file, and can be modified by the user to create nested floating menus visible during the simulation (fig4). Our configuration was specific for the CFD features that were also implemented in LS-PrePost 4.0.

Figure 3

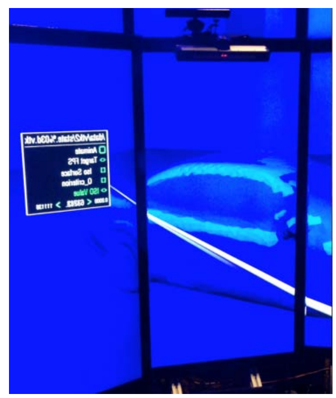

Figure 4

Workflow

The workflow processes for the CAVE and High Dynamic Range images and animations uses the same result files from LS-DYNA. Creating common geometry, surfaces and meshes allowed local coordinate systems and units of measure to work more efficiently between the software tools. This initially provided a struggling point for creating a seamless workflow. As an example, if the streamline results were post processed separately from the model, they needed combined and scaled appropriately before the rendering process.

Rendering for High Dynamic Range images and animations was defined with templates that included: predefined materials, lighting, rigging, and basic scene setup. Parameter changes were made by the user to vary these inputs with an add-on module inside Blender. An alternate method was to use a configuration file passed to the rendering nodes at runtime. Once the model template and job scheduler input is completed, each node begins an independent process to extract surfaces, apply template parameters, render assigned scenes, and return results to a network assigned storage device. The head node of the job uses an epilogue file to complete the final assembly of the images and animations in the project directory.

The CAVE output is rendered on the fly from the LS-DYNA results. A discretized set of data is generated from the ISO surface values to the polygons. Rendering usually take approximately 1-2 seconds to transverse the vector fields. Data sets with large values can be pre calculated to speed up the rendering time. CalVR by design runs a separate thread at a constant pull rate of 60Hz for the tracking subsystem. Because of this intentional design, slower frame rates for large data sets do not inhibit the performance speed. The binary output becomes the interactive 3D immersive environment as seen by the users on the combined panel system.

Because the CalVR framework works with passive displays, it is not limited to a single Head Mounted Display (HMD) and user. Each panel is independently driven by one of the NVIDIA GTX 1080 display adapters, and provides the same performance for all the user's point of view.

The independent panels can be configured to provide 120-degree field of view for the user by creating a concave semi-circle. Creating an immersive system such as this helps maintain the Plausibility and Place Illusion, thus convincing the user and their senses that they are actually there[11]. Our goal was to make sure the user focused on the data for analysis, and not the 3D space or objects in their peripheral view. Additionally, there is no need to consider the body illusion as with the HDM because the passive 3D glasses allow the users to see their self and others. This configuration is more ideal for the multi-user collaboration.

Data

The following results are from a rendered vehicle simulation with streamlines using available hardware. The output was generated for static images and animations files. Computation and rendering times were dependent on the hardware's capability to perform the assigned tasks. The investigation was focused on defining the methodology, and performance was not optimized in this example. We did complete a comparison between CPU and GPU calculation times, which increased the performance significantly.

The conversion of the streamlines to a particle emitter system for the animations is also included in the calculation and rendering times. We created photo realistic materials in the render to best reflect the real world lighting. These dielectric materials are also included in the rendered files and animations calculation times (fig5).

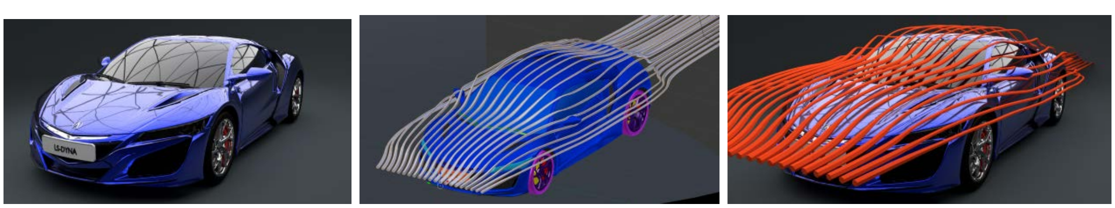

Figure 5

HDR Images and animations:

*Table 1: Rendered Data (CPU only). *Note: 1 VTK state file = 1 render frame.*

File size (Static/Animation)	**Computation time (250 frames)**
PNG image: 3.4MB	55 min/frame (average)
MP4 movie: 24MB	2 min, MP4, H.264 codec, 1920x1280

*Table 2: Rendered Data (GPU with CUDA libraries). *Note: 1 VTK state file = 1 render frame.*

File size (Static/Animation)	**Computation time (250 frames)**
PNG image: 3.4MB	10 min/frame (average)
MP4 movie: 24MB	2 min, MP4, H.264 codec, 1920x1280

CAVE Binary Output:

Table 3: CAVE data (Extracted VTK states).

File size (Binary/Meta)	**Computation time (250 states)**
.bin image file: 534MB	1 sec/frame or less at runtime.
. meta file: 19MB	N/A

Progress

We have implemented methods to view the new CFD capabilities for post processing using LS-PrePost and the CAVE. Feature extractions capabilities for transient CFD simulations were derived from research made available by the Massachusetts Institute of Technology[9].   This library now gives us the capability to go beyond a general set of tools, and utilizes additional important flow features such as vortex cores, separation/line attachment, vortices and line integral convolution. High Dynamic Range images and animations now include CFD results, where previously only structural models were used for research and testing. The workflow methodology for the investigation is moderately robust, but the choice to standardize on an enterprise platform in a clustered environment needs be considered further.

Coupled LS-DYNA results currently work with the immersion CAVE, and can be displayed independently or combined at runtime.   Our implementation of cutting planes in the CAVE visualizes real-time results as the user positions the cutting plane along the selected axis. This feature was not available in any other immersive CFD environment at the time of development, and works with all of the new supported features.

Future work

In the future, we want to explore combining the HDR models in a virtual reality environment with CFD and structural analysis results. The goal is to combine the best of both solutions into a single output for the analyst, designer, and decision maker. Focusing on open source tools, we tested the Blender Game Engine and the Unity platform during our investigation. We have concluded that Unity Engine is the best method for advancing our initial progress. It demonstrates capabilities for using the models, materials and lighting directly from our render process. Methods that reduce our need to build from scratch are more feasible given that we can identify all the unknown factors ahead of time. The Unity integration with the CalVR framework currently works for Windows based environments. The developers are working to also support the Linux operating system in the near future. Additionally, CalVR already has support for smart phones, tablets and other interactive devices such as Microsoft Kinect displaying its adaptability for new technology. The Unity platform builds for mobile, HMD and PC devices, and works with most operating systems. This combined pathway will be the future of our investigation as we try to develop the best method to attain our initial goal.

Summary

We presented our initial investigation for a proposed method to post process LS-DYNA results supporting new CFD functionality. The CFD capabilities are now available in LSPP 4.x for 2D analysis, and in our immersion CAVE for collaborative 3D analysis. We described the hardware, software and methods that met our needs to minimize our costs, and used available Open Source tools whenever possible. Selecting a passive display framework for our 3D CAVE system provided user control, and clearly demonstrated advantages for working in collaboration. Limiting the Virtual Reality environment to a single subject wearing an active VR Head

Mounted Display did not work with our process because multi user interaction was a requirement. To achieve HDR models of higher quality requires the use of physically correct shaders, materials and lighting. These requirements were beyond the capability of any available finite elements post processors during our investigation. We were also trying to take advantage of GPU processor capabilities for rendering, and tools that were already available from the Open Source community. We achieved a high rate of success in part to working with our customers, who provided valuable feedback and direction during the development process. The demand for using Virtual Reality in analysis has been increasing for the last fifteen years. We believe this trend will continue to increase, as physical prototypes will be replaced with virtual simulations especially in the automotive industry. Our initial investigation has provided adequate results, and allowing us to continue to develop more robust tools and solutions using an immersive system for analysis.

References

[1] Schulze, J., Prudhomme, A., Weber, P., Defanti, T., “CalVR: An Open Source Virtual Reality Software Framework”, (2013), 1-5.
[2] Sherman, B., “FreeVR: Virtual Reality Integration Library, http://www.freevr.org/,” (2008).
[3] OSG (OpenSceneGraph), “OSG home page, http://www.openscenegraph.org, “ (2017).
[4] LS-PrePost, “LS-PrePost support page, http://www.lstc.com/lspp/content/other/scripto/,” (2009).
[5] DeFanti, T., et all, “The Future of the CAVE, Central European Journal of Engineering,”(2010), 6,7,8,12,16,17,18.
[6] ParaView, “Paraview home page, http://www.paraview.org/,” (2017).
[7] Blender, “Blender home page, http://www.blender.org/”, (2017).
[8] Hess, R., “Blender Foundations”, Focal Press (2010), 331-362.
[9] Haimes, R., “Automated Feature Extraction form Transient CFD Simulations”, (2010).
[10] Haimes, R., Darmofal, D.,”Visualization in Computational Fluid Dynamics: A Case Study”, (1991), 1-5.
[11] Jaron Lanier, Mar Gonzalez-Franco, “Model of Illusions and Virtual Reality”, (2017), 1-6.