LAWRENCE LIVERMORE NATIONAL LABORATORY

LLNL-TR-741087

ATDM Rover Milestone Report STDA02-1 (FY2017 Q4)

M. Larsen, D. E. Laney

November 3, 2017

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

ATDM Rover Milestone Report STDA02-1 (FY2017 Q4)

Matt Larsen, Dan Laney

October 19, 2017

1 Introduction

We have successfully completed the MS-4/Y1 Milestone STDA02-1 for the Rover Project. This document describes the milestone and provides an overview of the technical details and artifacts of the milestone.

2 Milestone: [MS-4/Y1] Design and Implementation of Hardware accelerated volume rendering capability for simulated radiography diagnostics and visualization

Milestone Description: this milestone is focused on building a GPU accelerated ray tracing package capable of doing multi-group radiography, both back-lit and with self-emission as well as serving as a volume rendering plot in VisIt and other VTK-based visualization tools. The long term goal is a package with in-situ capability, but for this first version integration into VisIt is the primary goal.

Milestone Execution Plan:

- Create API for GPU Raytracer that supports multi-group transport (up to hundreds of groups).
- Implement components into one or more of: VTK-m, VisIt, and a new library/package implementation to be hosted on LLNL Bitbucket (initially), before releasing to the wider community.

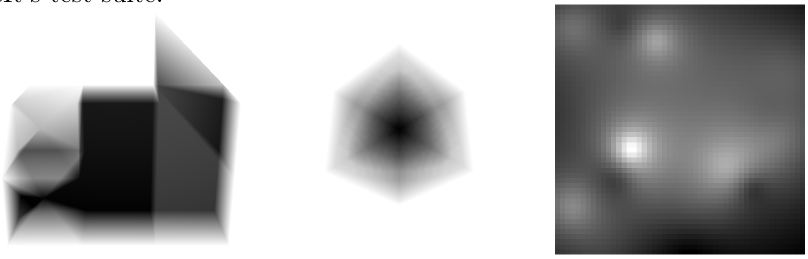
3 Technical Discussion

3.1 Create API for GPU Raytracer that supports multi-group transport (up to hundreds of groups)

We have created our alpha version of Rover which is a multi-group transport ray tracer capable of running on both CPU and GPU architectures. Our current test suite is capable of using a user-defined number of groups to enable testing of common use cases. Additionally, we have an opacity database with approximately 30 materials to simulate realistic conditions in our test suite. Figure 1 shows several examples of test problems within our framework.

The implementation effort was split into two components. Low-level ray tracing infrastructure was implemented using VTK-m. It is responsible creating acceleration structures and tracing rays in a shared-memory setting. We support absorption only, absorption plus emission, and volume rendering modes. The ray tracing infrastructure has been contributed back to the main VTK-m repository. The high-level component is responsible for the front-facing API, coordinating distributed-memory parallel compositing based on DIY [PRK+11], and ray generation.

Figure 1: Radiographs of three test data sets. On the left, an small mesh consisting of unstructured zoo elements. In the middle, the same unstructured data set from Figure ??. On the right, a data set used in VisIt's test suite.



3.2 Implement components into one or more of: VTK-m, VisIt, and a new library/package implementation to be hosted on LLNL Bitbucket (initially), before releasing to the wider community

The minimalistic API is documented with numerous unit test in the Rover repository located at LLNL's internal Bitbucket repository (https://lc.llnl.gov/bitbucket/scm/vis/rover.git). In addition to the test suite base on Google's gtest framework, we provide an example integration with a synthetic distributed-memory proxy application. Further, we provide a simple example of how to integrate with Rover's CMake build system. Rover's low-level components have been integrated into the ECP VTK-m library at https://gitlab.kitware.com/vtk/vtk-m, and the current VisIt [CBW+12] integration is located in a branch off the main trunk at http://visit.ilight.com/svn/visit/branches/mlarsen/rover2/.

4 Artifacts

Artifacts included in the completion of this milestone are as follows:

1. Ray tracing infrastructure located in the ECP VTK-m repository https://gitlab.kitware.com/vtk/vtk-m.
2. The Rover ray tracing library at https://lc.llnl.gov/bitbucket/scm/vis/rover.git (Internal to LLNL pending sofware review and release)
3. VisIt integration at http://visit.ilight.com/svn/visit/branches/mlarsen/rover2/

5 Acknowledgments

This work was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration. This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

References

[CBW+12] Hank Childs, Eric Brugger, Brad Whitlock, Jeremy Meredith, Sean Ahern, David Pugmire, Kathleen Biagas, Mark Miller, Cyrus Harrison, Gunther H. Weber, Hari Krishnan, Thomas Fogal, Allen Sanderson, Christoph Garth, E. Wes Bethel, David Camp, Oliver Rübel, Marc Durant, Jean M. Favre, and Paul Navrátil. VisIt: An End-User Tool For Visualizing and Analyzing Very Large Data. In *High Performance Visualization–Enabling Extreme-Scale Scientific Insight*, pages 357–372. Oct 2012.

[PRK+11] Tom Peterka, Robert Ross, Wesley Kendall, Attila Gyulassy, Valerio Pascucci, Han-Wei Shen, Teng-Yok Lee, and Abon Chaudhuri. Scalable parallel building blocks for custom data analysis. In *Proceedings of Large Data Analysis and Visualization Symposium LDAV'11*, Providence, RI, 2011.