Milestone Completion Report

**WBS 1.3.5.05 ECP/VTK-m
FY17Q3 [MS-17/02] Faceted Surface Normals, STDA05-3**

**Kenneth Moreland
Sandia National Laboratories**

July 11, 2017

EXECUTIVE SUMMARY

The FY17Q3 milestone of the ECP/VTK-m project includes the completion of a VTK-m filter that computes normal vectors for surfaces. Normal vectors are those that point perpendicular to the surface and are an important direction when rendering the surface. The implementation includes the parallel algorithm itself, a filter module to simplify integrating it into other software, and documentation in the *VTK-m Users' Guide*.

With the completion of this milestone, we are able to necessary information to rendering systems to provide appropriate shading of surfaces. This milestone also feeds into subsequent milestones that progressively improve the approximation of surface direction.

1. INTRODUCTION

This report documents the completion of the milestone "FY17Q3 [MS-17/02] Faceted Surface Normals" (listed as epic STDA05-3 in JIRA) for the ECP/VTK-m project. The overarching goal of the ECP/VTK-m project is to enable scientific visualization on the emerging processors required for the latest generation of petascale computers and the extreme scale computers of the future.

One of the biggest recent changes in high-performance computing is the increasing use of accelerators. Accelerators contain processing cores that independently are inferior to a core in a typical CPU, but these cores are replicated and grouped such that their aggregate execution provides a very high computation rate at a much lower power. Current and future CPU processors also require much more explicit parallelism. Each successive version of the hardware packs more cores into each processor, and technologies like hyperthreading and vector operations require even more parallel processing to leverage each core's full potential.

VTK-m is a toolkit of scientific visualization algorithms for emerging processor architectures. VTK-m supports the fine-grained concurrency for data analysis and visualization algorithms required to drive extreme scale computing by providing abstract models for data and execution that can be applied to a variety of algorithms across many different processor architectures.

Although there will be some time spent in the VTK-m project building up the infrastructure, the majority of the work is in redeveloping, implementing, and supporting necessary visualization algorithms in the new system. We plan to leverage a significant amount of visualization software for the exascale, but there is still a large base of complex, computationally intensive algorithms built over the last two decades that need to be redesigned for advanced architectures. Although VTK-m simplifies the design, development, and implementation of such algorithms, updating the many critical scientific visualization algorithms in use today requires significant investment. And, of course, all this new software needs to be hardened for production, which adds a significant overhead to development.

Our proposed effort will in turn impact key scientific visualization tools. Up until now, these tools — ParaView, VisIt, and their in situ forms — have been underpinned by the Visualization ToolKit (VTK) library. VTK-m builds on the VTK effort, with the "-m" referring to many-core capability. The VTK-m name was selected to evoke what VTK has delivered: a high-quality library with rich functionality and production software engineering practices, enabling impact for many diverse user communities. Further, VTK-m is being developed by some of the same people who built VTK, including Kitware, Inc., which is the home to VTK (and other product lines). Developers of ParaView and VisIt are in the process of integrating VTK-m, using funding coming from SciDAC and ASC. However, while VTK-m has made great strides in recent years, it is missing myriad algorithms needed to be successful within the ECP. Developing those algorithms is the focus of this ECP proposal.

2. MILESTONE OVERVIEW

The "FY17Q3 [MS-17/02] Faceted Surface Normals" milestone produces new functionality within VTK-m that introduces the ability to compute normal vectors, which are perpendicular to a surface they describe.

Surface normals are a critical component for lighting calculations that provide the major visual cues for surface shape. The first approximation for the normal to a surface represented by a polygonal mesh is the perpendicular direction to each flat polygon. This is simple to calculate, but provides a faceted representation of the surface.

This milestone produces an implementation of a surface normal worklet and filter that are merged to master branch of the central VTK-m repository. Documentation of the surface normal filter is added to the VTK-m User's Guide working document.

3. TECHNICAL WORK SCOPE, APPROACH, RESULTS

Computing the normal to a flat polygon is a relatively straightforward operation. The approach is to pick 2 edges of the polygon, define vectors parallel to these edges, and take the cross product of these 2 edges. The cross product result is then rescaled to be of unit length.

Computing the normal for a single polygon in a mesh representation of a surface can be done independently of any other normal computations. Such an operation is easily constructed within VTK-m using a so called point-to-cell map. In this construction, the computation for normal calculation with all associated data is independently executed on many concurrent threads.

The result of this operation is a single normal value for each polygon. Rendering polygons with this single normal value produces a flat appearance that highlights the polygonal facets that are approximating the surface.

4. CONCLUSIONS AND FUTURE WORK

The faceted appearance of surfaces is sometimes useful, but often we wish to approximate smooth shading. Future ECP/VTK-m will build on this functionality to produce interpolated normals that replicate the behavior of a curved surface.

ACKNOWLEDGMENTS