Geometric Algebra Computing

13.11.2014

Dr.-Ing. Dietmar Hildenbrand
Technische Universität Darmstadt

HSA foundation

TECHNISCHE UNIVERSITÄT DARMSTADT

AMD Announces Heterogeneous C++ AMP Language for Developers

August 26, 2014

First Open Source C++ Implementation to See Broad Availability Across Linux, Windows and Other Platforms SUNNYVALE, ...

C++ AMP (HSA foundation)

- in collaboration with Microsoft
- open source C++ compiler
- available on both Linux and Windows for the first time
- supporting three outputs:
  - OpenCL(1), supporting AMD CPU/APU/GPU, Intel CPU/APU, NVIDIA GPU, Apple Mac OS X and other OpenCL compliant platforms;
  - Khronos Group SPIR, supporting AMD CPU/APU/GPU, Intel CPU/APU and future SPIR compliant platforms;
  - HSA Foundation HSAIL, supporting AMD APU and future HSA compliant platforms.
- support for shared physical memory, which greatly simplifies sharing of data between the CPU and GPU on heterogeneous platforms

GA direkt in Programmiersprache?

TECHNISCHE UNIVERSITÄT DARMSTADT

Integrating Language Support for Geometric Algebra Computing into Scala

Bachelor Thesis, Master Thesis

GA computing is a new way of geometrically intuitive computing. Based on the underlying Geometric Algebra (GA), it is very easy to compute with geometric objects such as spheres, planes and circles or to handle geometric operations and transformations. For example, the intersection of two spheres, which is a circle, can be simply expressed based on the outer product of the algebra. This technology leads to more compact algorithms for many engineering areas such as computer graphics, computer vision and robotics as well as for simulations, e.g. molecular dynamics simulation.

The goal of this thesis is to build support for GA computing into Scala using SugarScala. SugarScala is a syntactically extensible variant of Scala where developers can introduce new language features through libraries. Using this technology, this thesis will design and integrate language constructs specific to GA computing. From here, there are two alternative directions:

- Use lightweight modular staging to compile GA programs to highly-efficient code (the optimizations are well-known), and (time permitting) use Delite to run GA programs in parallel.
- Design and implement a language on top of the SugarScala GA features that more easily allows experts from computer graphics, computer vision, robotics, or molecular dynamics to express GA programs.

For further information contact Sebastian Erdweg. This thesis will be co-supervised by Dietmar Hildenbrand.

Publications

- Erdweg, Rendel, Kästner, Ostermann: PDF SugarJ: Library-based syntactic language extensibility. In OOPSLA, 2011.
- Hildenbrand: Foundations of Geometric Algebra Computing.
- Charrier, Hildenbrand: PDF Geometric Algebra enhanced Precompiler for C++ and OpenCL.
- Lightweight modular staging.
- Delite.

Thanks a lot

Google Dietmar Hildenbrand