VISUALIZE YOUR LARGE DATASETS!

Peter Messmer, 3/20/2019
Extract information, gain insight
Visual cues, interactivity enhance focus
Helps to understand data
ParaView, VisIt, Matlab, Python, ...

Tell a story
Support story with visual FX
Catch viewer’s attention
Houdini, Blender, Maya, ..
VISUALIZATION ≠ RENDERING *

* but it’s a part of it

- Isosurfaces, Isovolumes
- Field Operators (Gradient, Curl,.. )
- Streamlines
- Coordinate transformations
- Feature extraction
- Clip, Slice
- Compositing
- Surface Rendering
- Thresholding
- Binning, Resample
- Line Rendering
- Volume Rendering
CHALLENGES AT LARGE SCALE

Locality

Complexity

Tools
CHALLENGES AT LARGE SCALE

Locality
Leave it where it is

Complexity
Use optimal resource

Tools
Minimal intrusion
VISUALIZATION IN THE DATACENTER
VISUALIZATION IN THE DATACENTER

Benefits of Rendering on Supercomputer

- Scale with Simulation
  - No Need to Scale Separate Vis Cluster

- Cheaper Infrastructure
  - All Heavy Lifting Performed on the Server

- Interactive High-Fidelity Rendering
  - Improves Perception and Scientific Insight
CHALLENGES IN THE DATACENTER

Headless rendering

Remoting

Vis Software Stack
HEADLESS RENDERING
HEADLESS RENDERING
How to rasterize without an attached display

OpenGL context management

Two approaches for context handling:
- X server: mgmt. by separate process
- EGL: mgmt. by driver
X SERVER ON HEADLESS
How to rasterize without an attached display

Recommended if code modification is not an option

nvidia-xconfig -o xorg.conf --allow-empty-initial-configuration -a

-o output file
-a enables all GPUs (--enable-all-gpus)
--allow-empty-initial-configuration start even if no attached display detected
CONTEXT MANAGEMENT WITH EGL

How to rasterize without an attached display

Requires minor application modification of GLX context initialization

```c
// 1. Initialize EGL
EGLDisplay eglDpy = eglGetDisplay(EGL_DEFAULT_DISPLAY);
EGLint major, minor;
eglInitialize(eglDpy, &major, &minor);

// 2. Select an appropriate configuration
EGLint numConfigs; EGLConfig eglCfg;
eglChooseConfig(eglDpy, configAttribs, &eglCfg, 1, &numConfigs);

// 3. Create a surface
EGLSurface eglSurf = eglCreatePbufferSurface(eglDpy, eglCfg, pbufferAttribs);

// 4. Bind the API
eglBindAPI(EGL_OPENGL_API);
```

https://devblogs.nvidia.com/egl-eye-opengl-visualization-without-x-server/
REMOTING
FLEXIBLE GPU ACCELERATION ARCHITECTURE

Independent CUDA Cores & Video Engines

Decode HW*

- Formats:
  - MPEG-2
  - VC1
  - VP8
  - VP9
  - H.264
  - H.265
  - Lossless

- Bit depth:
  - 8 bit
  - 10 bit
  - 12 bit

- Color**
  - YUV 4:2:0
  - YUV 4:4:4

- Resolution
  - Up to 8K***

Encode HW*

- Formats:
  - H.264
  - H.265
  - Lossless

- Bit depth:
  - 8 bit
  - 10 bit

- Color**
  - YUV 4:4:4
  - YUV 4:2:0

- Resolution
  - Up to 8K***

* Diagram represents support for the NVIDIA Turing GPU family
** 4:2:2 is not natively supported on HW
*** Support is codec dependent
VIDEO CODEC SDK
APIs For Hardware Accelerated Video Encode/Decode

What’s New with Turing GPUs and Video Codec SDK 9.0

• Up to 3x decode throughput with multiple decoders on professional cards (Quadro & Tesla)
• Higher quality encoding - H.264 & H.265
• Higher encoding efficiency (15% lower bitrate than Pascal)
• HEVC B-frames support
• HEVC 4:4:4 decoding support

NVIDIA GeForce Now is made possible by leveraging NVENC in the datacenter and streaming the result to end clients

**NVPIPE**

A Lightweight Video Codec SDK Wrapper

Simple C API

H.264, HEVC

RGBA32, uint4, uint8, uint16

Lossy, Lossless

Host/Device memory, OpenGL textures/PBOs

https://github.com/NVIDIA/NvPipe

Issues? Suggestions? Feedback welcome!
EGL RENDERING + BROWSER STREAMING

Powerful combo for rendering in the cloud

Available upon request

OpenGL Application | GLUT | EGLUT | NVENC | Web Server

mp4 stream

Java Script Client
TOOL COMPLEXITY
KITWARE PARAVIEW
Open-Source (Distributed) Visualization Package

OpenGL

NVIDIA IndeX Plugin
VTK: VISUALIZATION TOOLKIT
Open Source Scientific Visualization Toolbox

Process data using pipelines made up of filters

Forms the foundation of ParaView, VisIt and many other vis tools

OpenGL, Software raytracing

S9458 - VTK-m: Lessons from Building a Visualization Toolkit for Massively Threaded Architectures, Wed 3/20, 3:00-3:50
CONTAINERS: SIMPLIFYING WORKFLOWS

WHY CONTAINERS

Simplifies Deployments
- Eliminates complex, time-consuming builds and installs

Get started in minutes
- Simply Pull & Run the app

Portable
- Deploy across various environments, from test to production with minimal changes

S9525 - Containers Democratize HPC, Tue 3/19
NGC CONTAINERS: ACCELERATING WORKFLOWS

WHY CONTAINERS

Simplifies Deployments
- Eliminates complex, time-consuming builds and installs

Get started in minutes
- Simply Pull & Run the app

Portable
- Deploy across various environments, from test to production with minimal changes

WHY NGC CONTAINERS

Optimized for Performance
- Monthly DL container releases offer latest features and superior performance on NVIDIA GPUs

Scalable Performance
- Supports multi-GPU & multi-node systems for scale-up & scale-out environments

Designed for Enterprise & HPC environments
- Supports Docker & Singularity runtimes

Run Anywhere
- Pascal/Volta/Turing-powered NVIDIA DGX, PCs, workstations, servers and top cloud platforms
GPU-OPTIMIZED SOFTWARE CONTAINERS

Over 50 Containers on NGC

DEEP LEARNING
TensorFlow | PyTorch | more

MACHINE LEARNING
RAPIDS | H2O | more

INFEERENCE
TensorRT | DeepStream | more

HPC
NAMD | GROMACS | more

GENOMICS
Parabricks

VISUALIZATION
ParaView | IndeX | more
RENDERING: 2D
GPU ACCELERATED VECTOR GRAPHICS

Acceleration of 2D Graphics

GPUs primary rendering focus on 3D
2D rendering is so much more common
Often served out via web pages

Examples

graphs, diagrams, networks, flow charts, maps, vector artwork, Flash-like animation, etc. etc.
SCALABLE VECTOR GRAPHICS (SVG)

Pros:
- Wide support, efficient implementations
- Very powerful feature set

Cons:
- Slow due to client-side rendering in browser
- SVG contains data, not just pixels

⇒ GPU cloud rendering addresses both downsides
⇒ Support via NV_path_rendering OpenGL extension
SVG RENDERING PERFORMANCE

Bigger benefit for more complex scenes

 Quadro RTX6000
i7-5820k
ANATOMY OF A RAY-TRACING APP

Interplay of Rays and Geometry

- Intersection of rays with geometry
- Arbitrary new rays started at arbitrary locations
- Arbitrary operations at intersection points
- Typically in 3D space
- Hierarchical spatial decomposition as acceleration structure
TURING RT CORES

Hardware Accelerated Ray Tracing

RT Cores perform

- Ray-BVH Traversal
- Instancing: 1 Level
- Ray-Triangle Intersection

Return to SM for

- Multi-level Instancing
- Custom Intersection
- Shading

Programming via OptiX RT framework
Low overhead interop with CUDA

S9768 - New Features in OptiX 6.0
Wed 3/20, 1:00-1:50pm
BETTER INSIGHT VIA RAYTRACING

It’s not just pretty pictures
OPTIX AI DENOISER IN PARAVIEW

Without Denoiser

With Denoiser
VISRTX
Visualization Framework Powered by NVIDIA RTX Technology

Progressive forward pathtracer with NEE/MIS

Hardware-acceleration through OptiX

MDL for physically-based materials

AI denoiser

Area lights, Depth of Field, Tone mapping, etc.

Open-source C++ library

Feedback welcome (issues, PRs, e-mail)!

http://github.com/NVIDIA/VisRTX
VISRTX + PARAVIEW

VisRTX open-source on GitHub

Shipped with upcoming ParaView 5.7

- No additional steps necessary!
RAYTRACING PAYS OFF AT SCALE

ParaView Manyspheres Benchmark
RENDERING: VOLUMES
NVIDIA IndeX SDK

Large scale and distributed data rendering
Scene management with volume data
Transparent support for NVLink
Higher-order filtering, advanced lighting & transfer functions

https://developer.nvidia.com/index
NVIDIA INDEX FOR PARAVIEW PLUGIN

- NVIDIA IndeX rendering in ParaView
- Retain ParaView workflows
- Structured and unstructured meshes

Learn more:
SUMMARY

Wide Palette for Visualization and Rendering in Datacenter/Cloud

Headless rendering

Accelerated video streaming

2D graphs can benefit from GPUs as well

Raytracing great to enhance vis perception

VisRTX raytracing vis toolkit (in ParaView, VTK)

GPU accelerated scalable volume rendering part of open source tools