Interactive Processing and Visualization of Geospatial Data

Dr. Brian Hendee Smith
Technical Fellow, Boeing Advanced Network & Space Systems
March 26, 2014
Geospatial Data Processing

- Large volumes of data collected from air borne and space borne sensors
- Processing extremely computationally intensive
  - Traditionally minutes to hours timelines
  - User may need to iterate/optimize algorithms to get usable result
  - Results may need to be transferred to/from analyst location

How can GPUs change how we analyze geospatial data?
Geospatial Sensor Data Processing

Collection Platforms
- Target Area

Raw Data
- Electro-Optic Imagery
- Synthetic Aperture Radar
- Hyper-spectral

Processing
- Transform data to human viewable format

Analyst
- Engineer
- Scientist
- Domain Expert
Examples Data Sets

Synthetic Aperture Radar (SAR)
- Digitized Receive Pulses
- Magnitude/Phase Data
- May include multiple polarizations

- Image Formation (Tomographic-like reconstructions)
- Polarimetric Analysis
- Interferometric Analysis

Hyper-Spectral
- Band Selections
- Principal Component Analysis
- Material Detection

λ – 250 samples
y – 900 samples
x – 1800 samples
Interactive Processing

Boeing Advanced Network & Space Systems

- Raw Sensor Data
  - Unprocessed Data is loaded onto video card

- Data Processed on GPU
  - Results rendered directly to display via OpenGL
  - Operator adjusts algorithm parameter via GUI controls
  - Algorithm parameters updated and data reprocessed at 10-40 frames/second

On-Card Feedback Loop Enables Instantaneous Operator Feedback
Early Prototype

- Early demonstration based on OpenGL examples in CUDA SDK
- Demonstrated utility of + interactive processing
Agility Framework provides interactive processing for geospatial analyst
- Leverage GPU on analyst’s workstation
- Near instantaneous feedback to algorithm change
- Flexibility to redefine algorithmic processing chain on-the-fly
- Interoperability with remote batch processing

Agility Framework provides interactive processing for algorithm developers
- Run-time addition of algorithms
- Allows change to algorithm parameters
- Modular support for differing sensor types
- Separate concerns of processing device from signal/image processing logic
Agility Framework

Processing Modules & Sensor Interfaces Provisioned at Run-time
Agility Interactive Processing Demonstration

MiniSAR Imagery Courtesy of Sandia National Lab
Agility SARLab


- SAR phase history processing and rendering at 10-20 fps
- Rapid and comprehensive exploitation of SAR phenomena and production of actionable information
- Simple, yet powerful, analysis workflow
- Inexpensive, small footprint supercomputing capability for SAR processing and exploitation
- 10-100x processing speed improvement over traditional SAR processing approaches
- Supporting SAR data formats including TerraSAR-X, Radarsat-2, COSMO-SkyMed, Sensor-Independent SICD, and several other widely used formats

MiniSAR Imagery Courtesy of Sandia National Lab
GOTCHA Imagery Courtesy of Air Force Research Labs
Interactive Algorithm Editing

- Integrated algorithm development toolkit
- Developers can edit Python scripts to define algorithm modules
- Resulting algorithms run immediately on GPU as part of workflow
Agility Web Services

Sensor Data → Agility Servers → Network → Agility Web Client

Fast end-to-end delivery
Agility Web Services

Boeing Advanced Network & Space Systems

HTTP Request
- Data Set
- Algorithms

Results
- JPEG
- NITF
- KML
- Metadata

AGILITY VIRTUAL MACHINES
- HS Algorithms
  - GPU

Computer Infrastructure
Summary

Direct interaction with data

- Dynamic API for sensor format definitions
- Python based dynamic interface for defining GPU processing algorithms