Initial work on development of an open Streaming Media Standard for Field of Light Displays (SMFoLD)

Jamison R. Daniel, Benjamın Hernandez,
C.E. Thomas Jr, Steve L. Kelley, Paul G. Jones,
Chris Chinnock

Third Dimension Technologies

Stereo Displays & Applications
January 29, 2018

Standard for Streaming 3D Media

- Sponsored by the Air Force (AFRL)
- Program Facilitators
  - Third Dimension Technologies
  - Oak Ridge National Laboratory
  - Insight Media

SMFOLD.ORG

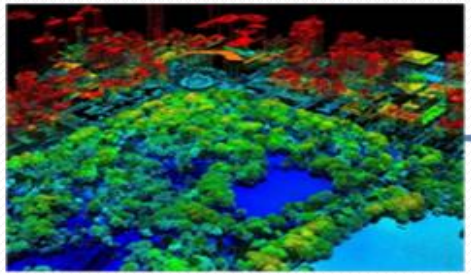
Motivation for SMFoLD

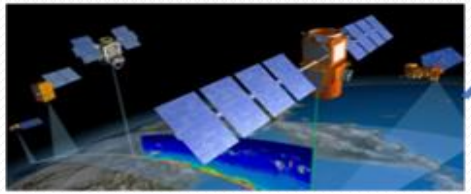
- 3D Data Increasing Dramatically
  - LiDAR, SAR, plenoptic camera, stereo or multi-view to 3D
  - 3D models (actual and created)
- 3D Visualization Needed to Improve Productivity
  - Stereoscopic 3D (S3D) not acceptable
  - Field of Light Display (FoLD) is desired
- Lack of Streaming Standard is Barrier to FoLD Adoption
  - Commercial standards bodies are focused on 2D and S3D (MPEG-I and JPEG Pleno are working on Light Field standards)
  - Common model needed for application portability
  - Nonproprietary standardized format for plug and play

Field of Light Displays (FoLD)

- Light field displays attempt to perfectly reproduce the actual 3D field of light that surrounds us
- See True 3D without glasses
- No gear strapped to your face
- No visual conflict discomfort
- Enhanced visual perception, situational awareness, sense of presence, and cognition

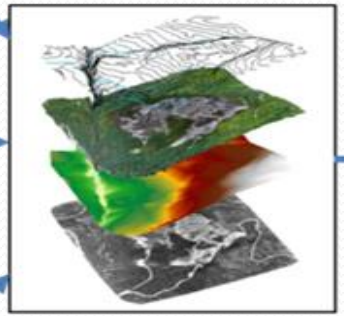
SMFoLD Use Case Example





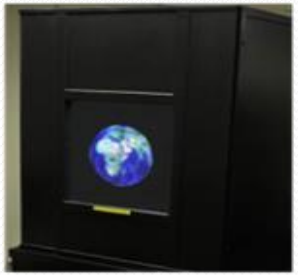
Enormous Amounts of 3D Data Being Generated

3D Streaming?



Spatially Heterogeneous and Multi-Scale Data with Inherently 3D Context

New Class of Field of Light Displays (FoLDs)



Third Dimension Technologies

- TDT is a Developer of FoLD Systems
- Well Versed in Multiview Rendering Challenges
- Current Projects
  - FoLD Integrated Flight Simulator
  - Standard for Streaming 3D Media to Field of Light Displays (SMFoLD)

3RD DIMENSION TECHNOLOGIES

Problem to be Solved

- FoLD Systems Must Render 3D Scenes from Multiple Viewpoints
  - FoLDs generate gigapixels, streaming not feasible
  - 3D applications should not be tied to a specific display type
  - FoLDs need access to shaders
  - S3D applications typically provide only one viewpoint

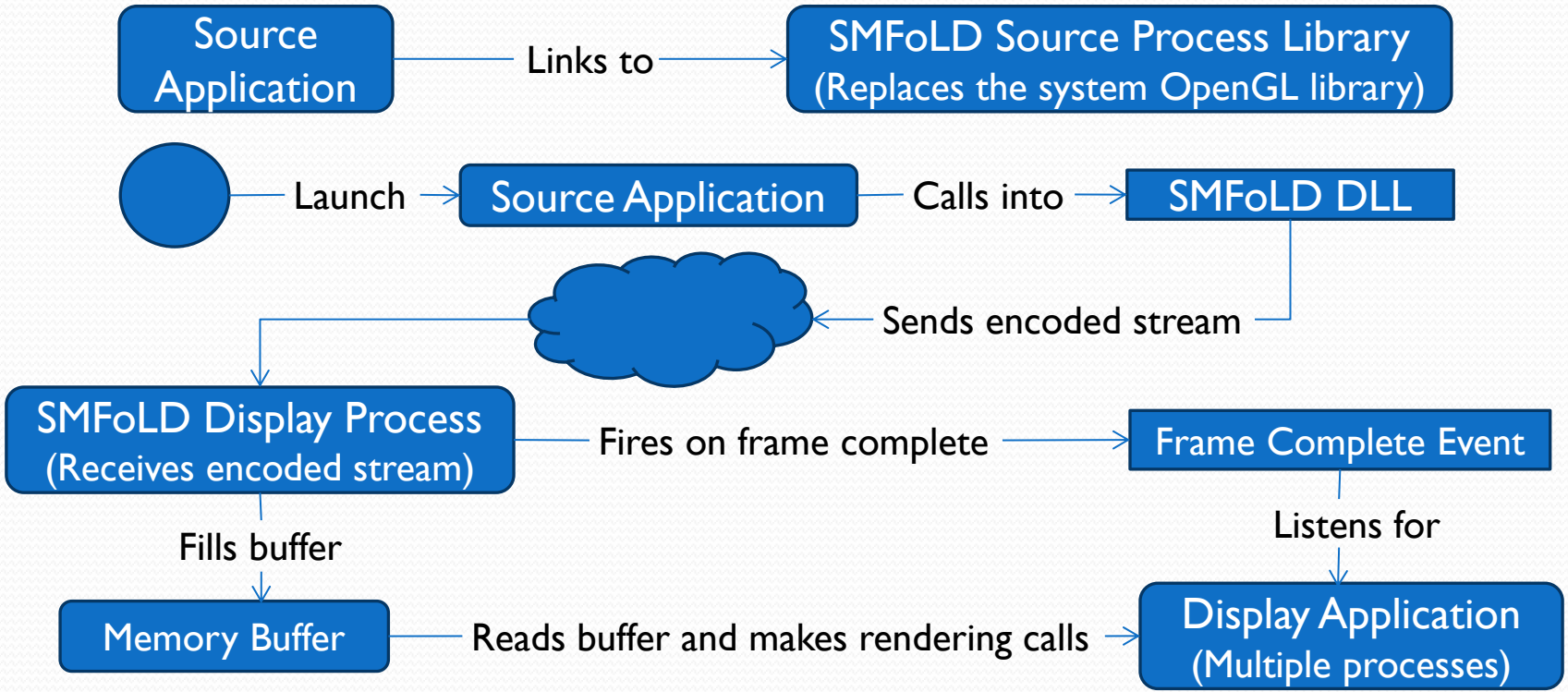
SMFoLD Objectives

- Display Agnostic 3D Streaming
  - Viewable on any 2D, S3D or FoLD system
  - Includes scene description and transmission format
  - Allows for flow and point of view control
- Open International Standard
- Supports Commercial and DoD Needs
- Supported by Display and Application Developers
- Allows More 3D Applications to Run on FoLD Systems

SMFoLD Approach

- Use OpenGL for the SMFoLD Interface
  - Source Applications link to SMFoLD library (SMFoLD.lib)
  - Display Applications include SMFoLD definitions
- Data Types Limited to Mesh, Texture & OpenGL Primitives
  - Mesh formats reduce bandwidth requirements
- Hooks to Named Variables ("Uniforms") in Shaders
  - Camera position, camera angle, focal plane, camera field of view, and others (TBD)

SMFoLD Flow Model



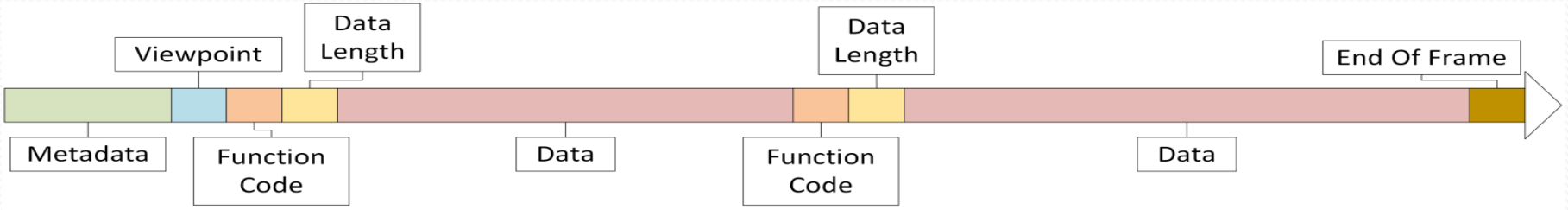
What is in an SMFoLD 3D Frame?

- All Information Needed to Display Image in 2D or 3D
  - Values that represent function calls (opcodes)
  - Data that the function calls use (function arguments)
  - Metadata to allow the display to create any number of viewpoints
- 3D Objects Defined as Discreet Objects
  - Object definition downloaded once and stored locally
  - Object can be locally manipulated without changing the object data

What is in an SMFoLD 3D Frame? (2)

- Graphics Primitives (functions) Used to Describe 3D Scene
  - Viewpoint as defined by the application
  - Metadata needed by 3D displays for multi-viewpoint rendering
  - Geometry transformation matrix
  - Colors, material properties, blending, etc.
  - Arrays of values expressing 3D structures or models
- Shaders - rendering pipeline logic for all frames

Typical SMFoLD 3D Data Frame Format

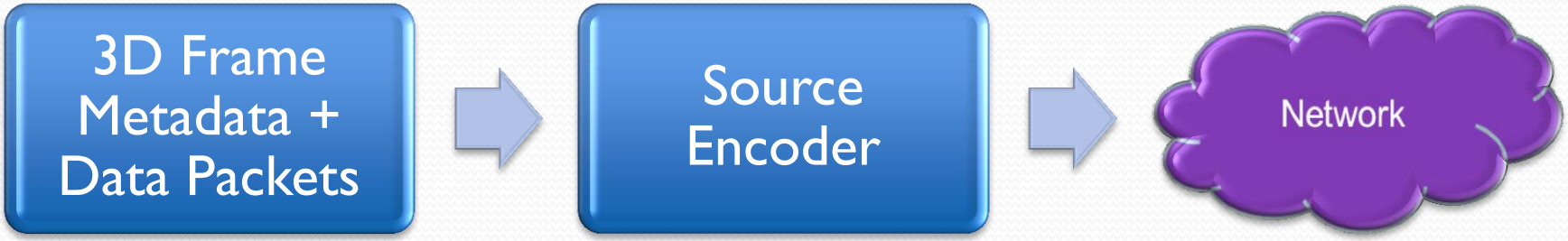


Structure of the SMFoLD Stream

Source Process Encodes 3D Frame

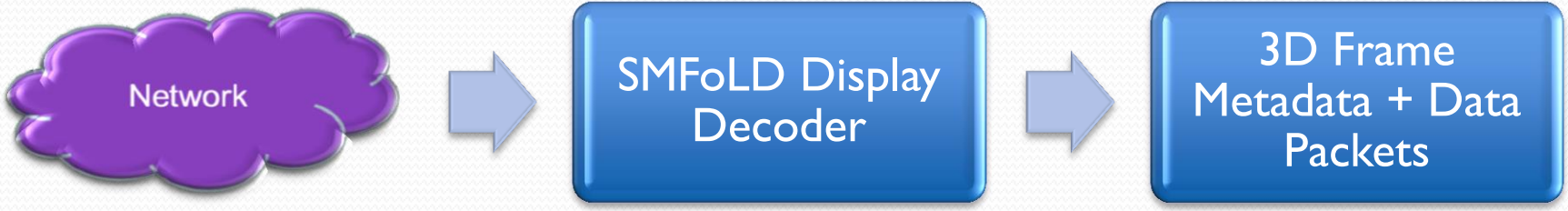
- Application Processes 3D Data
- Application Calls Source Process API to Create 3D Frame
- Source Process Encodes Functions and Arguments
- Writes Encoded Data to Memory Buffer

3D Frame Prepared for Transmission



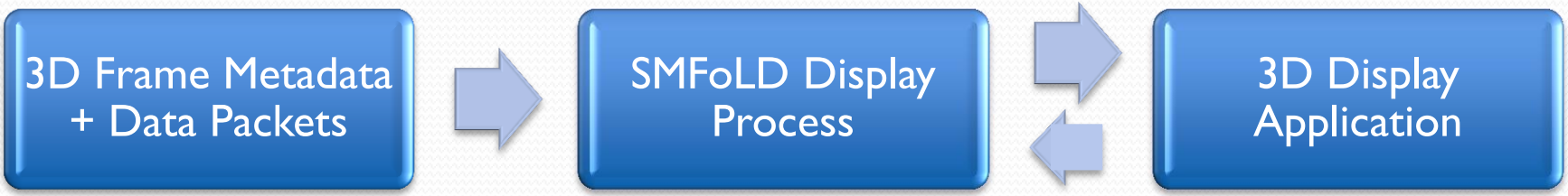
- 3D Frames Compressed and Encrypted
- 3D Frames Output to Network

3D Frame Decoded



- 3D Frames Decompressed and Decrypted
- 3D Frame Restored to Opcodes and Data

Display Creates GPU Code



- Display Process Writes Incoming Stream to Memory
  - Notifies Display Application when frame is ready for rendering
- Display Application Performs Multiview Rendering
- Metadata Used to Set Display Specific Parameters
- Shaders
  - Shader hooks used to set view geometry

Shaders

- Graphics Pipeline is Implemented in Shaders
- Vertex Shader Performs Geometric Transformations
- Shader Hooks Allow Geometry Changes by Display Application
  - Hooks are named variables that a display application can access
  - Source application shader code is required to provide hooks
- SMFoLD Defined Shader Function Called by Vertex Shader
  - Function will use the named variables

Conclusions

- OpenGL API Graphics Primitives Plus Extensions Provides a Short Path to Display Agnostic 3D Streaming
- High Frame Rates can be Achieved Over Existing Networks
- Different 3D Data Types can be Added
- Need Support of Application and Display System Providers

Next Steps

- Implement Model and Share Test Results
- When Authorized Share as Open Source Software
- Engage Application and Display Providers
- Attend SMFoLD Workshops
- Visit SMFoLD.org
- Find a Home With an Existing Standards Body