AFRL-RH-WP-TR-2019-0098

Streaming Model for Field of Light Displays (SMFoLD) Phase II

C. E. Thomas Jr*, Del R. Barstow*, Chris W. Honsinger*, Steve L. Kelley*, Paul G. Jones*, Andrew G. Smith*, Jamison R. Daniel, Ben Hernandez Arreguin**, and Chris Chinnock*****

***Third Dimension Technologies LLC, 11020 Solway School Rd, Knoxville TN 37931**
****Oak Ridge National Laboratory, 1 Bethel Valley Rd, Oak Ridge TN 37831**
***** Insight Media, 3 Morgan Ave, Norwalk CT 06851**

December 2019
FINAL REPORT

THIS IS A SMALL BUSINESS TEHCNOLOGY TRANSFER (STTR) PHASE II REPORT

DISTRIBUTION STATEMENT A: Approved for public release: distribution unlimited.

AIR FORCE RESEARCH LABORATORY
711 HUMAN PERFORMANCE WING
AIRMAN SYSTEMS DIRECTORATE
WRIGHT-PATTERSON AIR FORCE BASE OH 45433
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

NOTICE AND SIGNATURE PAGE

AFRL-RH-WP-TR-2019-0098 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

//signature//
DARREL G. HOPPER, Ph.D., DR-IV
Work Unit Manager, Sensory Systems Branch
Airman Systems Directorate
711 Human Performance Wing
Air Force Research Laboratory

//signature//
LOUISE A. CARTER, Ph.D., DR-IV
Chief, Warfighter Interface Division
Airman Systems Directorate
711 Human Performance Wing
Air Force Research Laboratory

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
21-12-19	Final	19 Sep 2017– 18 Dec 2019

4. TITLE AND SUBTITLE

Streaming Model for Field of Light Displays (SMFoLD) Phase II

5a. CONTRACT NUMBER FA8650-17-C-6877

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER 65502F

6. AUTHOR(S)

C. E. Thomas Jr*, Del R. Barstow*, Chris W. Honsinger*, Steve L. Kelley*, Paul G. Jones*, Andrew G. Smith*, Jamison R. Daniel, Ben Hernandez Arreguin**, and Chris Chinnock*****

5d. PROJECT NUMBER 3005

5e. TASK NUMBER V0

5f. WORK UNIT NUMBER H0T3 (Historic: 3005V003)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

*Third Dimension Technologies Inc., 11020 Solway School Rd Ste 104, Knoxville TN 37931
**Oak Ridge National Laboratory, 1 Bethel Valley Rd, Oak Ridge TN 37831
*** Insight Media, 3 Morgan Ave, Norwalk CT 06851

8. PERFORMING ORGANIZATION REPORT NUMBER TDT-AFRL-20191221

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Material Command
Air Force Research Laboratory
711 Human Performance Wing, Airman Systems Directorate
Warfighter Interface Division, Sensory Systems Branch
Wright-Patterson Air Force Base OH 45433

10. SPONSORING/MONITORING AGENCY ACRONYM(S) USAF AFMC 711 HPW/RHCS

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S AFRL-RH-WP-TR-2019-0098

12. DISTRIBUTION/AVAILABILITY STATEMENT DISTRIBUTION STATEMENT A: Approved for Public Release; distribution unlimited.

13. SUPPLEMENTARY NOTES
This is a Small Business Innovative Research (SBIR) Phase I report developed under a contract awarded under STTR topic AF16A-T07 Streaming Model for Field of Light Displays (SMFoLD). Contains Patentable Information (patent being pursued but not approved). 88ABW Cleared 02/07/2020; 88ABW-2020-0419.

14. ABSTRACT Report developed under STTR topic AF16A-T07. The lack of a standard model for streaming 3D data has been identified as an impediment to the use of Field of Light Display (FoLD) systems. The FoLD class comprises several types, including voxel-based volumetric, hogel-based light field, and plenoptic projection. The plenoptic projection FoLD type is an inverse-plenoptic camera with its array of image sensors replaced by an array of image projectors of overlapping viewpoints. Third Dimension Technologies (TDT) with sponsorship from the Air Force Research Laboratory (AFRL) has collaborated with Oak Ridge National Laboratory (ORNL) and Insight Media (IM) to develop a standard for streaming 3D visuals. The streaming model is display agnostic and can nominally be used on any SMFoLD compliant display (FoLD, stereo, or 2D). The SMFoLD model has been implemented with the creation of SMFoLD Source and Application Dynamic Link Libraries (DLLs): "*SMFoLD_Source.lib*" and "*SMFoLD_Display.lib*". Demo Source and Display Applications have also been written, and SMFoLD has been demonstrated on two (2) different types of FoLD system: (1) the LightSpace x1406C, 20 LCD depth screen full parallax (FP) volumetric type and (2) the TDT LaunchTN 22-channel horizontal parallax only (HPO) plenoptic projection type. The LightSpace demo was held 1 Aug 2019 during the contract final review meeting hosted by AFRL at Wright-Patterson AFB OH, and on various other occasions at the facilities of TDT The TDT LaunchTN demo was completed on 6 Dec 2019 at the facilities of TDT.

15. SUBJECT TERMS
Field of Light Display, FoLD, Streaming Model, SMFoLD, Sensory Systems, Battlespace Visualization, Air-Space-Cyber Battle Management Center, Full Parallax, True 3D , Holography, Lightfield, Volumetric, Plenoptic Projection

16a. REPORT	16b. ABSTRACT	16c. THIS PAGE	17. LIMITATION OF ABSTRACT:	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON (Monitor)	19b. TELEPHONE NUMBER *(Include Area Code)*
Unclassified	Unclassified	Unclassified	SAR	149	Dr. Darrel G. Hopper	(937) 255-8822

TABLE OF CONTENTS

Section **Page**

LIST OF FIGURES

LIST OF TABLES

Table **Page**

ACKNOWLEDGMENTS

Third Dimension Technologies LLC (TDT) would like to thank the Air Force team for their interest in this project and their selection to fund this Small Business Technology Transfer (STTR) Phase II effort. In particular, TDT would like to thank:

- Dr. Darrel Hopper, Air Force Research Laboratory (AFRL)
- Dr. Fred Meyer, AFRL
- Eric Heft, AFRL
- Alexander Enzmann, MITRE

Additionally, TDT would like to thank TDT team members and other individuals that contributed to the success of this feasibility project. The list below briefly summarizes the individuals and companies that provided important technical input based on discussions, feedback, guidance and/or data:

- Jamison Daniel, Oak Ridge National Laboratory
- Ben Hernandez, Oak Ridge National Laboratory
- Chris Chinnock, Insight Media

FOREWORD

The PE65502F $749,881.00 STTR Phase II contract FA8650-17-C-6877, AFRL Workunit H0T3, Job Order Number (JON) (3005V003), was awarded to Third Dimension Technologies LLC (TDT) on 19 Sept 2017 with a base period end date of 18 Dec 2019. TDT teamed with the Oak Ridge National Laboratory (ORNL) as the STTR-required research institution, and with Insight Media to support communications and the workshops required by the effort. The Phase I purchase order FA8650-16-M-6750, AFRL WU (JON) H0Q8 (3005CV46) in the amount of $149,995.00, was awarded to TDT on 28 Jul 2016 and ended on 1 Apr 2017. The Phase I Final Report has been published to the Defense Technical Information Center (DTIC); the citation is:

> C. E. Thomas Jr, Steve L. Kelley, Paul G. Jones, A. Smith, Jamison R. Daniel, Ben Hernandez Arreguin, and Chris Chinnock
> "*Open Standard for Display Agnostic 3D Streaming (DA3DS) Phase I*"
> AFRL-RH-WP-TR-2017-0029, Third Dimension Technologies LLC, Knoxville TN, 90 pp (April 2017). Distribution B. Available to qualified requesters at www.dtic.mil

These efforts were selected/awarded under the DoD BAA 16.A STTR Topic program entitled

"AF16A-T07 Streaming Model for Field of Light Displays (SMFoLD)"

sponsored by the AFRL Airman Systems Directorate, 711 HPW/RHCS (pka RHCV). The OBJECTIVE, DESCRIPTION, PHASE I goals, and PHASE II goals of this topic are as follows.

OBJECTIVE:
Develop a model for the multimedia data stream required for next generation Feld of Light Display (FoLD) systems to project full-parallax video-rate 3D images without eyewear. Demonstrate the model on a FoLD system in a command center environment.

DESCRIPTION:
Collection, storage, transmission, and viewing of 3D data by a variety of DoD sensor systems has increased dramatically over the past 15 years and even more rapid growth is anticipated. The Air Force Life Cycle Management Center (AFLCMC) Battle Management Directorate has identified a Technology Need for true 3D visualization systems to increase productivity of operators dealing with the 3D data deluge. AFLCMC further requires the data be viewed without special eyewear on a new class of display, a so-called FoLD visualization system.

A variety of prototype FoLD systems have been developed that each uses a unique, proprietary approach to transmit and visualize the same 3D data. The government (Defense Advanced Research Projects, DARPA, Intelligence Advanced Research Projects Agency, IARPA, Air Force) has sponsored several efforts to foster the development of FoLD systems. Each effort has recreated the underlining software to ingest 3D content for delivery to their device. Lack of a common streaming media model has emerged as a barrier creating FoLD systems acceptable within a command center environment.

Government leadership is required. The focus of commercial standards bodies has been exclusively on the Stereo 3D (S3D) class of 3D display. The S3D class requires special eyewear and is, for a variety of reasons, not acceptable in a command center environment. S3D has caused eye fatigue and nausea in certain viewers due to a conflict in the accommodation and vergence cues it provides to the human visual system. The nausea can be reduced, but not eliminated, if the viewer is stationary and the content is tailored pixel by pixel (which is possible in movies over several months of post-production but wholly impractical in a command center). Furthermore, S3D has limited value for parallax correct viewing since the perspectives are simulated from imagery that was captured from only one or two points of view (POV). These human interface limitations of S3D have prevented its adoption to address the 3D data deluge in Air Force command centers.

The emerging new FoLD class of 3D visualization system offers non-eyewear full parallax viewing and perspectively correct visualization for multiple persons. The FoLD class comprises several types including lenticular, volumetric, and holographic. Furthermore, many existing 3D capture methodologies based on Light Detection and Ranging (LiDAR) sensors, Synthetic Aperture Radar (SAR) sensors, or plenoptic cameras capture a 3D environment that can be viewed correctly from many perspectives only on a FoLD visualization system.

Today the burden on integrating a FoLD system into an application space or environment is placed, over and over, on each software application developer. The emerging hardware technologies have yet to unite behind a common model for streaming a 3D scene description. Proprietary 3D display hardware and software formats limit the adoption and interchange of 3D visualization devices.

The next step in the evolution of 3D visualization is the creation of a common Streaming Model for 3D data--including a scene description protocol and transmission format--that is display technology agnostic. The standard should define a streaming 3D scene that can be viewed on any 2D, S3D or FoLD visualization system and allow such flow and POV control as is required by the host application or content. Current and future display prototypes in any class (FoLD, S3D, and 2D) could then create an optimal visualization from the same streaming scene description.

PHASE I:
Define display-technology agnostic, 3D streaming model for FoLD systems that is similar to existing 2D protocols. Establish definitions for streaming 3D content, audio content, compression, metadata, encryption, key frames, and error recovery. Integrate protocol and definitions into the model. Organize and conduct workshop open to all government and industry to publicize results.

PHASE II:
Revise Streaming Model to address industry comments at the workshop and publish as a technical report to be entitled "Draft Data Streaming Model for Field-of-Light Display (FoLD) Visualization Systems." Brief the report at multiple scientific and engineering meetings including SMPTE, IEEE, and SID. Conduct a second workshop and revise the technical report. Document performance tradeoff analysis of choices made in a final report. Develop a software tool to implement the model.

1.0 SUMMARY

This report documents TDT's development efforts for the SMFoLD standard from ~19 Sept 2017 to 18 Dec 2019 on a Phase II STTR contract for AFRL at Wright-Patterson Air Force Base (WPAFB).

TDT, in collaboration with ORNL, has developed a conceptual design for a display agnostic standard for streaming 3D graphics known as SMFoLD (Streaming Model for Field of Light Displays). The proposed standard will allow SMFoLD compliant displays that can produce 3D graphics to receive a stream of 3D frame descriptions and render a 3D scene. The computer driving the display device must be running an SMFoLD compliant display application. Illustrated in Figure 1 below is a typical command center application in which an application receives streams of 3D data from multiple sensor sources, fuses the data into a 3D graphical narrative, and then streams the resultant 3D imagery to multiple display types.

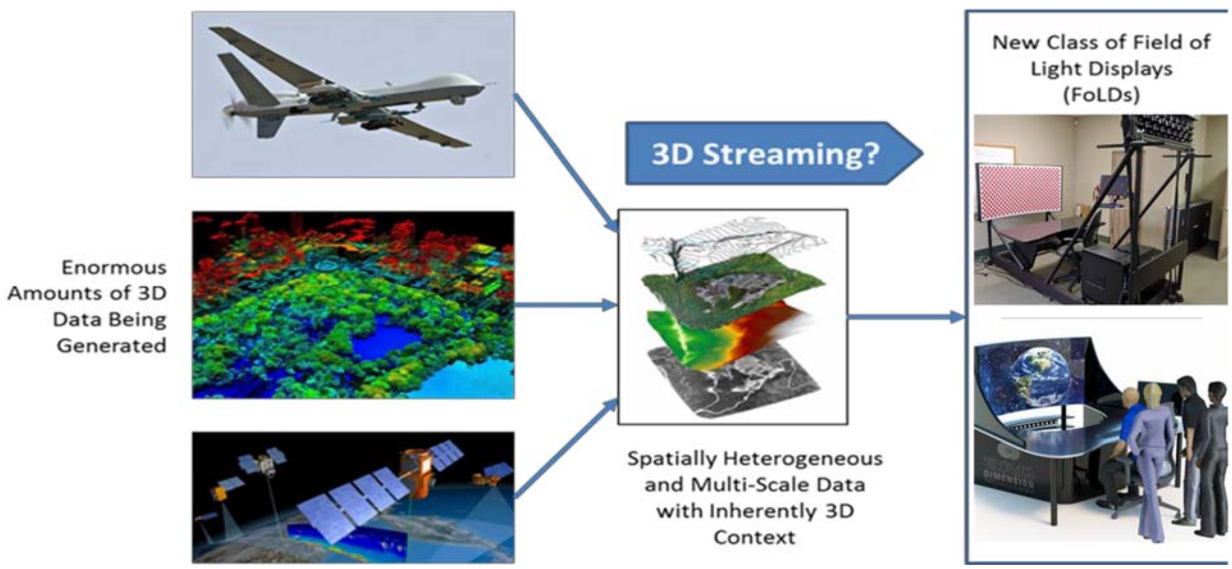

Figure 1. Example of Command Center 3D Data Streaming Application

1.1 Planned Work and Goals

The scope and tasks to be performed, as defined in the Contract non-proprietary Statement of Work (SOW) are listed below.

The scope of this effort is to:

(a) develop a Streaming Model for Field of Light Display systems;

(b) develop all software required to demonstrate an implementation of the SMFoLD open standard;

(c) test the performance of the standard; and

(d) promote the standard in order to gain support for adoption.

Non-Proprietary Statement of Work

Task 1.0: **Draft SMFoLD Standard/Technical Report (TDT/ORNL)**

The contractor, TDT, and its STTR research partner subcontractor, ORNL, shall create and document a draft streaming model and standard architecture capable of supporting all types of FoLD and S3D visualization systems. Industry feedback shall be incorporated in the Technical Report.

Task 2.0: **SMFoLD Phase II Workshop (IM/TDT/ORNL)**

The contractor, along with its subcontractors Insight Media LLC (IM) and ORNL, shall organize and conduct a Phase II workshop to publicize the streaming model and draft standard to industry. The contractor shall obtain industry feedback for incorporation into the SMFoLD standard.

Task 3.0 SMFoLD Standard Implementation

Task 3.1 SMFoLD Software Architecture (TDT/ORNL)

- The contractor shall create the architecture for the SMFoLD standard and outline a software tool for its implementation.
- The contractor shall develop the SMFoLD Source Process.

Task 3.2 SMFoLD Display Process (ORNL/TDT) and Source Process (TDT/ORNL)

- The contractor shall develop the SMFoLD Display Process and Source Process.

Task 3.3 Test Code Applications (TDT/ORNL)

- The contractor shall develop the Test Source application

Task 3.4 Platform for Demonstration and Test (TDT/ORNL)

- The contractor shall implement the Demonstration & Test Platform on two different types of FoLD systems.

Task 3.5 Change Frame Capability (TDT/ORNL)

- The contractor shall add Change Frame Capability to the SMFoLD standard and software tool.

Task 3.6 Compression Capability (ORNL/TDT)

- The contractor shall add Compression capability to the SMFoLD standard and software tool.

Task 3.7 Encryption Capability (ORNL/TDT)

- The contractor shall add Encryption capability to the SMFoLD standard and software tool.

Task 3.8 Audio Capability (ORNL/TDT)

- The contractor shall add Audio capability to the SMFoLD standard and software tool.

Task 3.9 Display Application Feedback (TDT/ORNL)

- The contractor shall add Display Application Feedback capability to SMFoLD standard and software tool.

Task 4.0 Streaming Performance (TDT/ORNL)

- The contractor shall evaluate Streaming Performance of the SMFoLD software tool.

Task 5.0 Promote Adoption of the Standard (TDT/ORNL/IM)

- The contractor shall promote adoption of the SMFoLD and associated draft standard via publications and presentations at three or more technical conferences.

Task 6.0 Revised SMFoLD Standard (TDT/ORNL)

- The contractor shall update and revise the SMFoLD Standard drafted in Task 1.0 to incorporate lessons learned and feedback from Task 2.0 (Phase II workshop).

Task 7.0 SMFoLD Reference Code (TDT/ORNL)

- The contractor shall develop and publish the SMFoLD Reference Code software tool as open source.

Task 8.0 Explore Commercial Potential and Product Viability

- The contractor shall explore the potential to transition the SMFoLD and draft standard to professional standards development organizations, such as SMPTE (Society of Motion Picture and Television Engineers) and SID (Society for Information Display), for incorporation into their products.

Task 9.0 Manage Program and Submit Reports

- The Contractor shall exercise program management, administrative and financial management functions during the course of the program through reviews, teleconferences, reports, publications, and meetings, as required. The Contractor shall document performance tradeoff analysis of choices made in a final report.

Task 10. Operational Security (OPSEC)

- The contractor shall comply with general OPSEC procedures and apply them throughout the lifecycle of the contract. OPSEC procedures, policies, and awareness are required in an effort to reduce program vulnerability from successful adversary collection and exploitation of critical information.

1.2 Comparison of Achieved to Non Proprietary Statement of Work

This report documents TDT's development efforts for the SMFoLD standard from ~19 July 2017 to 18 Dec 2019 on a Phase 1I STTR contract for AFRL at WPAFB. The Phase I effort has led to the development of a conceptual design for a 3D graphics streaming protocol that is being developed and demonstrated in the Phase II project. This report discusses the methods and results for each task in the context of the main Phase II efforts with a brief summary of achievements as follows.

Task 1.0: Draft SMFoLD Standard/Technical Report (TDT/ORNL)

- A draft Technical Report entitled "Draft Data Streaming Model for Field-of-Light (FoLD) Display Visualization Systems" has been created and transmitted to AFRL on 18 Dec 2017. The Technical Report has been further revised and submitted again to AFRL with the Final Scientific and Technical Report in December 2019.

Task 2.0: SMFoLD Phase II Workshop (IM/TDT/ORNL)

- SMFoLD Workshop. TDT organized and hosted a workshop on 3 Oct 2017 where industry leaders presented their ideas and experience in the area of streaming video, 3D graphics, and standards development. An AFRL approved summary of the workshop along with presentations has been posted to the SMFoLD.org website. Another Workshop was held on 2 Oct 2018 in conjunction with the Display Summit 2018 Conference. Both of these workshops are documented in APPENDIX D.

Task 3.0 SMFoLD Standard Implementation

Task 3.1 SMFoLD Software Architecture (TDT/ORNL)

- The overall software architecture for SMFoLD is discussed in Section 4.3.

Task 3.2 SMFoLD Display Process (ORNL/TDT) and Source Process (TDT/ORNL)

- The SMFoLD Source Process (Source DLL) is discussed in Section 4.3.2.

- The SMFoLD Display Process (Display DLL) is discussed in Section 4.3.2.

Task 3.3 Test Code Applications (TDT/ORNL)

- The Test Source and Display Applications are both discussed in Section 4.3.2.

Task 3.4 Platform for Demonstration and Test (TDT/ORNL)

- The complete SMFoLD implementation was demonstrated on the LightSpace x1406C Volumetric Display at AFRL on 1 Aug 2019. SMFoLD was additionally demonstrated on the LaunchTN 22-channel FoLD system at TDT in Knoxville during December 2019.

Task 3.5 Change Frame Capability (TDT/ORNL)

- Change Frame capabilities are discussed in Section 4.3.4.

Task 3.6 Compression Capability (ORNL/TDT)

- Compression is discussed in Section 4.3.5.

Task 3.7 Encryption Capability (ORNL/TDT)

- Encryption is implemented using WebSockets and is discussed in Section 4.3.6

Task 3.8 Audio Capability (ORNL/TDT)

- TDT's current SMFoLD audio implementation does not work and has not been included in the present SMFoLD implementation. This is discussed further in Section 4.3.7.

Task 3.9 Display Application Feedback (TDT/ORNL)

- Display Application feedback is discussed in Section 4.3.8.

Task 4.0 Streaming Performance (TDT/ORNL)

- SMFoLD Streaming Performance is discussed in Section 4.3.9.

Task 5.0 Promote Adoption of the Standard (TDT/ORNL/IM)

- Presentations and publications about the SMFoLD standard are discussed in Section 1.3.

Task 6.0 Revised SMFoLD Standard (TDT/ORNL)

- The SMFoLD Technical Report has been revised and submitted to AFRL in December 2019.

Task 7.0 SMFoLD Reference Code (TDT/ORNL)

- The contractor has developed a private GitHub repository to publish the SMFoLD reference code. This repository will be converted to an open source public repository when approved by AFRL.

Task 8.0 Explore Commercial Potential and Product Viability

- The contractor has briefed MPEG, SMPTE, SID and JPEG members on the SMFoLD standard and has solicited their interest in the standard. This was accomplished at a number of conferences and in briefings and papers as reported in Section 1.3

Task 9.0 Manage Program and Submit Reports

- The contractor has submitted 24 monthly reports, conducted 14 Review Meetings and provided additional documentation and Draft and Final Scientific and Technical Reports.

Task 10. Operational Security (OPSEC)

- The contractor has made significant efforts to maintain Operational Security. Computers are all locked and protected, have antivirus programs, and are behind one or more network firewalls, both hardware and software.

Table 1 below shows the milestones for each task, the original planned date of completion, (month of project), and an actual date of completion when applicable.

Table 1. Key Milestones for TDT's SMFoLD Phase II Project

Milestone	Description	Month/Date
1	• Project Award • Project Kickoff Meeting • Monthly Status Report	19 Sept 2017 28 Sept 2017 18 Oct 2017
2	• Monthly Status Report • Installable SMFoLD Source Process implemented • Installable SMFoLD Display Process implemented	18 Nov 2017 18 Nov 2017 18 Nov 2017
3	• Monthly Status Report • Draft SMFoLD Standard Technical Report completed	18 Dec 2017
4	• Monthly Status Report • Source Application running	18 Jan 2018 12 Jan 2018
5	• Monthly Status Report • First SMFoLD Workshop Completed	19 Feb 2018 3 Oct 2017
6	• Monthly Status Report • Workshop report published • Display Application and rendering process running	18 Mar 2018 24 Nov 2017 18 Mar 2018
7	• Monthly Status Report	18 April 2018
8	• Monthly Status Report	18 May 2018
9	• Monthly Status Report • Test Plan (software test plan 9MAC, June 18, 2018)	18 June 2018 18 June 2018
10	• Monthly Status Report • Demonstration & test platform implemented	18 July 2018 6 Aug 2018
11	• Monthly Status Report	18 Aug 2018
12	• Monthly Status Report • Interim (Year 1) Phase II Summary Report (700 Words) • Demo at AFRL • Software Test Report (12MAC, 18 Sep 2018)	18 Sep 2018 Video Submit 18 Sep 2018
13	• Monthly Status Report	18 Oct 2018
14	• Monthly Status Report • Change Frame Capability implemented	18 Nov 2018 15 Dec 2018
15	• Monthly Status Report	18 Dec 2018
16	• Monthly Status Report • Compression implemented and tested	19 Jan 2019 18 Jan 2019
17	• Monthly Status Report • Second SMFoLD Workshop Completed	18 Feb 2019 2 Oct 2018

18	• Monthly Status Report • Encryption implemented • Second SMFoLD Workshop report published	18 Mar 2019 18 Mar 2019 18 Oct 2018
19	• Monthly Status Report	18 Apr 2019
20	• Monthly Status Report • Audio implemented • Display Application Feedback implemented	20 May 2019 Not Completed 25 July 2019
21	• Monthly Status Report • Software Test Plan (21MAC, 18 Jun 2019) • Software User Manual (Initial Submission 21 MAC, 18 Jun 2019)18	18 Jun 2019 18 Jun 2019 18Jun 2019
22	• Monthly Status Report • Streaming performance documented • SMFoLD standard tested and demonstrated on at least two different FoLD systems (nominally TDT HAS3D systems and the LightSpace multi-planar volumetric system)	18 July 2019 Dec 2019 LightSpace 1 Aug 2019 LaunchTN 6 Dec 2019
23	• Monthly Status Report	18 Aug 2019
24	• Monthly Status Report • Final (Year 2) Phase II Summary Report (700 Words) • Four papers presented at conferences • SMFoLD Standard Technical Report revised • Software User Manual (24MAC, 18 Sep 2019) • Submit Software Source Code and Executables. • SMFoLD implementation updated • Software Test Report (24MAC, 18 Sep 2019)	18 Sep 2019 18 Sep 2019 Sec. 1.3, Itm B Dec 2019 Dec 2019 Dec 2019 Dec 2019 Dec 2019
25	• Draft Final Phase II Scientific & Technical Report	Oct 2019
26	• Final Phase II Scientific & Technical Report	Dec 2019
The "Month" column denotes the month after the contract start date.		

1.3 Comparison of Achieved to SMFoLD Topic AF16-AT07 Required Performance

Completion dates for planned Phase II tasks are shown in Table 1 and this document is the Final Scientific and Technical Report. The Gantt chart in Figure 2 shows the planned tasks (blue bars) and percent completed (black bars within blue bars). All tasks have been completed except audio implementation.

Listed below are the requirements defined in the Phase II STTR topic: "AF16-AT07 Streaming Model for Field of Light Display (SMFoLD)." Listed under each requirement are the sections of this report that address that requirement.

A. Revise streaming model to address industry comments at the workshop and publish as a technical report to be entitled "Draft Data Streaming Model for Field-of-Light Display (FoLD) Visualization Systems".

(a) This requirement is addressed in Section 4.2 and subsections.

B. Brief the report at multiple scientific and engineering meetings, including SMPTE, IEEE, and SID.
   (a) The SMFoLD project was briefed at the SMFoLD workshop in Chantilly, VA, on 3 October 2017 (the SMFoLD workshop is discussed in D-1).[5]
   (b) The SMFoLD project was also briefed at the Display Summit Conference on 4 October 2017.[6]
   (c) The SMFoLD project was briefed at the SD&A conference on 29 January, 2018, and a paper was published. See Appendix Section B-1.[7]
   (d) The SMFoLD project was briefed at the Display Summit/SMFoLD workshop on 2 October 2018 (see Appendix D-2).[8]
   (e) An SMFoLD paper for has been approved by AFRL and submitted for publication to the SMPTE Motion Imaging Journal.[9]
   (f) An SMFoLD briefing has been presented at the 2019 Light Field and Holographic Display Summit (CableLabs, Louisville, CO, 9 Oct 2019 (see Appendix B-7).[10]
   (g) Charts for a SMPTE Technology Webcast have been approved by AFRL and the Webcast was presented on 14 Nov 2019.[11]

C. Conduct one or more additional workshops (two planned) and revise the technical report.
   (a) All of APPENDIX B addresses this requirement.
   (b) SectionD-1 discusses the second SMFoLD workshop (the first SMFoLD workshop was held during the SMFoLD Phase I STTR project and documented in the Final Report for that project).
   (c) A third SMFoLD workshop was held in conjunction with the Display Summit 2018 Conference on 2 October 2018 (see section D-2).

D. Document performance tradeoff analysis of choices made in a final report.
   (a) Sections 4.3.10 address this requirement.

E. Develop a software tool to implement the model.
   (a) Section 4.3 discusses the software development.

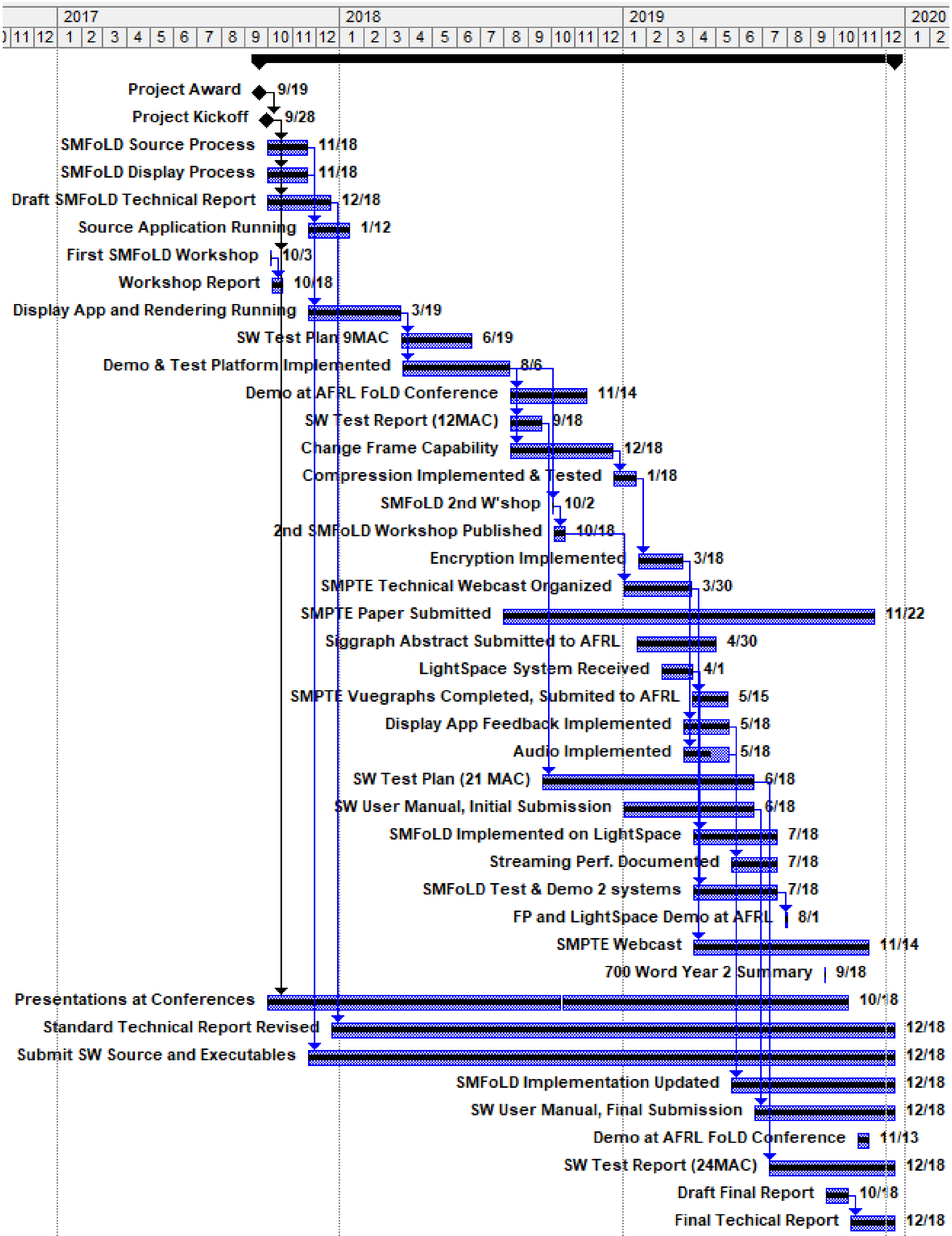

Figure 2. SMFoLD Project Gantt Chart - Planned vs Actual

2.0 INTRODUCTION

2.1 Air Force Need

The Air Force has identified a need for the creation of a common streaming model for 3D data—including a scene description protocol and transmission format—that is agnostic to the display technology at the end user.

Two factors are driving this need. The first is the dramatic increase in the collection, storage and transmission of 3D data from a wide array of Department of Defense (DoD) sensors (e.g., high-definition 2D and 3D imagery from manned/remote aircraft, satellites and other battlefield sensors). The second factor is that the Air Force has further identified a requirement for true 3D displays that do not require special eyewear to assist Warfighters with this deluge of 3D data (e.g., 3D terrain, multi-story buildings, synthetic aperture radar, LIDAR and 2D video).

In response to this second factor, a new class of 3D displays has emerged known as Field of Light Displays (FoLD). Unfortunately, today's FoLD systems lack a common 3D streaming model, and thus each FoLD system often implements a proprietary stovepipe streaming model, leading to one-time-use development. These ad hoc 3D streaming models are a barrier to the adoption of FoLD systems and the interchange of software among devices.

TDT has formed a consortium to address this need and to define a model based on open standards for Display Agnostic 3D Streaming (DA3DS). TDT and ORNL bring a wealth of 3D experience to this project including leadership in holographic 3D displays and considerable industry contacts with AMD and nVidia.

2.2 Technical Challenges and Opportunities

The scope of the 3D streaming problem is illustrated in Figure 3. Of particular note is the diversity and range of existing solutions and the lack of cohesion. The current state of 3D streaming is one that has yet to coalesce around a set of open standards as other media have done (e.g., MP3 for audio, JPEG for images and H.264 for video). The battle for a common 3D streaming model is currently being waged, and 3D is the last media without such a delivery format.

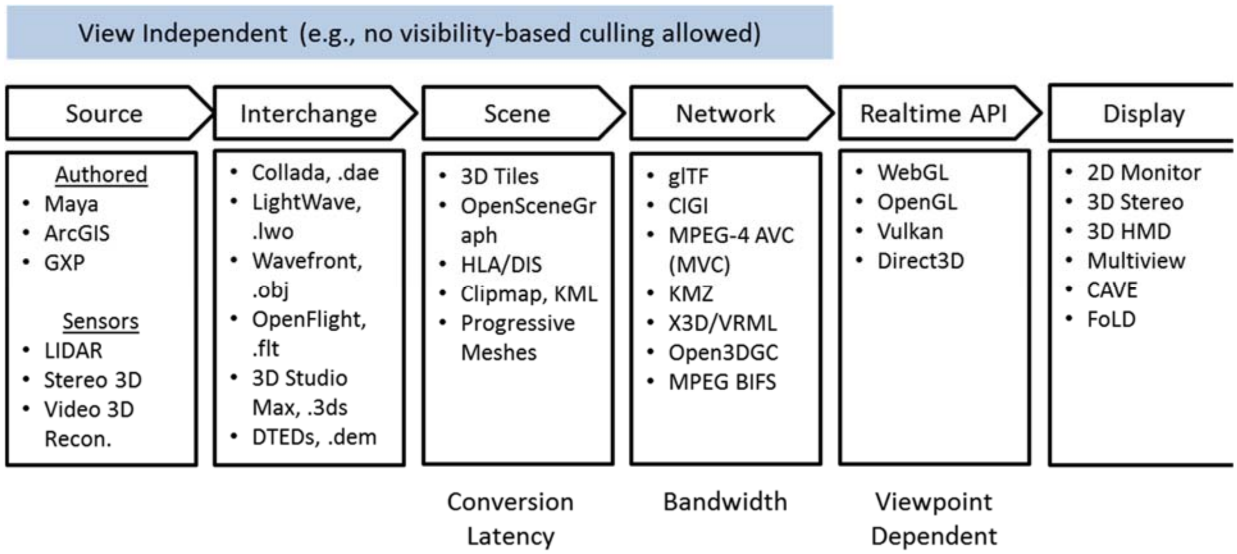

Figure 3. Block Diagram Illustration of SMFoLD Flow from Source to Sink

This block diagram begins with the Source, which can be a sensor or perhaps even a person (authored content), and progresses to the Display sink, which can be a traditional 2D, stereo 3D, multi-view 3D (e.g., lenticular) or a 3D FoLD display. Below each stage is a list of proprietary or open formats that are currently available for that stage. Notable examples include COLLADA as a file Interchange format and OpenGL for rendering via a real-time API. Less well defined stages are the Scene where OpenSceneGraph and 3D Tiles offer potential open solutions and the Network where glTF (gl Transmission Format) is a new open standard from the Khronos Group with interesting potential. Issues such as latency, bandwidth and view dependence appear under each stage as appropriate.

For the purposes of the SMFoLD Standard, the interchange format is an SMFoLD OpenGL Frame, as defined by SMFoLD Phase II, the scene is described by OpenGL mesh and texture plus metadata (point of view, POV, focal plane, field of view (FOV), other metadata as needed) and OpenGL primitives. The network transmission format is serialized packed and encoded OpenGL frames using WebSockets (TCP/IP), and the rendering API for the Display DLL is OpenGL (OpenGL opcodes defined by the SMFoLD project) which the DLL can translate to OpenGL and transmit to the Display Application.

2.2.1 Viewpoint Challenge

Among these issues, view dependency is one of the key challenges for 3D streaming to overcome. With traditional 2D displays, an inherent—and often unrecognized—assumption is that a single viewpoint with a particular rendering geometry defines the conversion of 3D data into a 2D image. This view dependency can appear throughout the various stages and subsequently breaks support for FoLD displays, which nominally have many viewpoints and not just one. Thus, viewpoint dependency introduces a contradiction for FoLD systems.

Ideally, a 3D streaming model would be viewpoint independent (e.g., no visibility culling allowed, avoid 2D constructs such as zoom level, etc.) from the Source through the Network until reaching the Realtime API stage. At this point, a calibration of the FoLD display would inject one or more viewpoints as needed to drive the FoLD system. This approach is a paradigm shift in 3D rendering concepts, and the SMFoLD standard proposes to resolve this problem by using the OpenGL view matrices to allow the Display Application to provide a different POV for each of its cameras (displays—hogels, projectors, diffractions sources, etc.). The particular FoLD receiving the stream must have its own calibration routines and provide its own POV for each camera. A sample FoLD Display Application and Calibration program are provided as demonstration samples with the SMFoLD source code and executables.

2.2.2 Streaming Challenge – Comparison with Media (Audio, Video & Image) Streaming

The development of a 3D streaming model is not new and has waxed and waned over the years:

(a) Beginning with early efforts on VRML in the 1990s,
(b) Continuing with simulation and training (S&T) efforts on DIS and CIGI in the 2000s, and
(c) More recently launching of gaming efforts with WebGL since 2009.

While other media formats such as MP3 audio, JPEG images and H.264 video are recognized delivery standards, 3D standards still remain elusive. While the analogy between 3D streaming and other media streaming is a tempting one, the comparison in Table 2 reveals differences that have stymied 3D standards from coalescing. Media (audio, image and video) streaming share an organic progression with a common lineage in signal processing theory, and thus are more amenable to sampling, compression and other signal elements. With 3D streaming, connections to signal processing are more ambiguous and less obvious. Thus, the world of 3D streaming is currently chaotic with a plethora of ad hoc formats and protocols. Recall Figure 3.

Table 2. Comparison of 3D Streaming to Media Streaming Models

	3D Streaming Realtime Graphics	**Media Streaming** Audio, Images and Video
Sampling	Non-uniform, irregular, aperiodic (point clouds, triangle meshes)	Uniform. Regular, periodic (image pixels, grids, bit streams)
Viewpoint	Locally dynamic (user can manipulate)* & back-channel (out of band) selection	Globally fixed (user cannot change); back-channel (out of band) selection
Data Structures	Multi-dimensional and varied (graphs, vectors, meshes, textures, points, manifolds), typically heterogeneous	Format fixed (matrix), primarily homogeneous
Resolution	Unbounded (models can be quite large)	Bounded (bit-depth and image size fixed by format)
Architecture	Client/Server (few, if any, Peer-to-Peer)	Mixture of Client/Server and Peer-to-Peer
Animation	Non-linear procedural (TRS – translate, rotate, scale); Also linear key frames possible, too.	Linear key frames (Phi Phenomenon)
Compression	Difficult to exploit redundancy across mixed heterogeneous data structures	Well-posed redundancy in space and time for signal theory

**See discussion of viewpoint challenges in previous subsection.*

The comparison in the table further illustrates the challenges with 3D streaming, mainly non-uniform sampling, heterogeneous data structures and unbounded resolution. These traits are in contrast to traditional media streaming models, which have stronger ties to signal processing. The lack of a strong signals foundation complicates efforts—particularly with compression—across the 3D data structures. Nominally, within each 3D streaming data structure, compression and signal elements are better defined, and so for highly focused efforts, such as for stereo or multi-view lenticular displays, standards such as the MPEG Multi-view Video Coding (MVC) standard with Stereo (e.g., 2D+Z or 2D+Delta) and Multiview Profiles[12] are likely more appropriate than the full 3D streaming model for FoLD systems discussed in this project. This project proposes a complete 3D streaming model that avoids possible shortcomings and artifacts from inherently image-based streaming models.[13]

2.3 State of the Art

The following discussions briefly review the state of the art for 3D streaming and FoLD systems, see Figure 4.

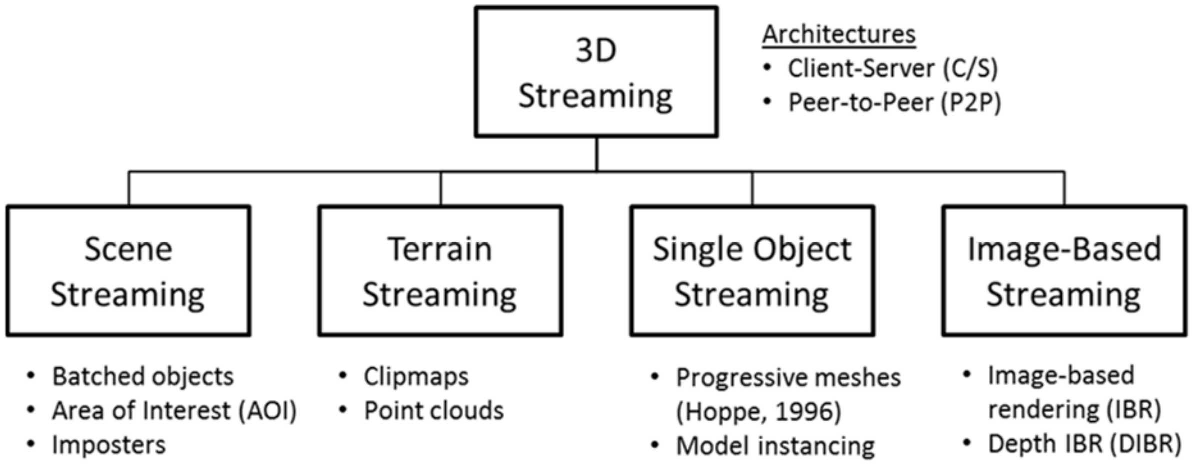

Figure 4. Categorization of State of Art 3D Streaming Efforts

2.3.1 3D Streaming

The diagram in Figure 4 shows a categorization of 3D streaming efforts. Scene streaming involves general techniques that transmit entire scenes each frame. Some scene techniques are essentially extension of rendering pipelines and include WireGL,[14] Chromium,[15] and similar methods,[16,17,18] including work by ORNL.[19] Scene methods also include peer-to-peer concepts[20,21] and more recently important efforts to create browser-level streaming with WebGL and glTF from the Khronos Group.[22,23] Google Earth first popularized terrain streaming with the important contribution of Clipmaps[24] with others contributing significant enhancements.[25] The work of Hoppe at Microsoft Research introduced the notion of object streaming with the seminal paper on Progressive Meshes.[26] The last streaming model is image based rendering (IBR) methods[27,28] with major efforts on Multi-view Video Coding (MVC) in MPEG standards.[12] TDT's experience suggests that IBR methods are not a general 3D solution to support the spectrum of FoLD systems.[29,13] However, the architectural framework of the MPEG committee (with notable absence of explicit viewpoint in the forward streaming protocols) is important and is similar to Figure 4. MVC MPEG and other 3D streaming protocols typically employ back-channel (out of band) transmission of viewpoint selection and explicit viewpoints in the forward channel or standard.[30,13] An emerging area that touches each of these categories is cloud-based mobile gaming,[31] which has important streaming lessons for user acceptance and adoption.[32]

For the SMFoLD project TDT has chosen to stream SMFoLD frames made up of metadata, OpenGL mesh and texture, and OpenGL primitives.

2.3.2 3D Display Systems

Over the last ~65 years, the work of D. Gabor[33] has inspired serious efforts in developing holographic 3D displays and associated field of light displays. See reviews.[34,35,36] True diffractive holographic displays (e.g., MIT Media Lab[37,38,39] and Univ. of Arizona[40]) are decades away due to pixel size and processing needs, [41,42] but their data requirements should be considered in the development of a new 3D streaming standard.

Aside from the technical challenges, diffractive holograms have far more information than is useful to human vision. So researchers long ago developed holographic stereograms,[43,44,45,46] which reproduce all human visual cues[47] with orders of magnitude greater efficiency than diffractive holography. TDT's HAS3D[48,49] is an electronic version of a holographic stereogram and thus eliminates eye fatigue and display sickness[47,36] and provides head motion parallax[50] (i.e. "look around" 3D viewing).

Other 3D display technologies include RealView[51] (note, videos of free-space "holograms" on the RealView website are computer-generated videos—not actual holograms), Holografika[52] and FoVI3D (formerly Zebra Imaging).[53] The FoVI3D Integral Ray or Hogel-based technology is of considerable interest as one of the leading full-parallax FoLD technologies under development. New versions of the FoVI3D technology are expected to be available over the next several years. SeeReal Technologies[54] is a European company with Headquarters in Belgium and further facilities in Germany. Their technology is based on narrow field of view actual diffractive holography, with head-tracking.

More recently Light Field Lab (LFL), has been developing an active pixel system with a wave guide lens to direct the pixels into a 3D light field.[55] LFL is targeting the cinema market where use of larger pixel sizes (say five microns?) substantially reduces the rendering workload that would be required for smaller pixels and large screens—the overall rendering workload. However, the rendering workload will be at least as large as the rendering workload for a smaller screen with the two micron pixels required for high resolution in a smaller integral ray display. Other recent entrants into FoLD systems development field include Holochip Corp. and Avalon Holographics Inc. Both companies appear to be developing hogel based FoLD systems, conceptually similar to and with the same computational burdens and pixel pitch constraints as the FoVI3D technology. Avalon's Holographic Rendering System, currently capable of processing 60 Gpixels/sec, may prove useful to FoLD developers in the near term.

Recently the Light Space volumetric 3D display has been resurrected (it was out of business for some period of time, and previously known as the Depth Cube). The display is based on a single-chip DLP engine in rear-projection monitor configuration. The imaging array consists of a series of addressable screens, one behind the other. By rapidly illumination one screen in its scattering configuration, with all the others set to transmissive, the display creates a 3D volumetric image.[56,57] LightSpace is continuing to actively develop the display.

XiGen[58] and Physical Optics Corp.[59] also appear to have developed 3D FoLD displays, having received several government research grants, but limited public information is available. Other non-holographic glasses-free 3D displays include lenticular (Philips WOWvx, Alioscopy, Zecotek) and parallax barrier (Sharp, Setred[60]) displays but these systems have many challenges.[47,35] Finally, some systems claim to be holographic[60,61] but are not.[62,34,36] Glasses based stereoscopic display systems (including movie theaters and Head Mounted Displays) are also available. These stereoscopic systems introduce vergence accommodation conflicts and make significant portions of the human population sick or uncomfortable.

2.4 TDT Flight Simulator Prototype and Future Workstation Concept

TDT has been developing FoLD systems since 2003. Figure 5 shows TDT's latest FoLD system (70" diagonal screen and 22 projector illumination) integrated in a flight simulator. TDT developed the simulator under an Air Force SBIR contract and delivered the system to AFRL in January, 2017. The FoLD display provides 3D visual cues required for training near object flight missions such as aerial refueling, formation flight, take-off and landing (in particular carrier landings), and close air support.

A concept for a Horizontal Parallax command center workstation based on TDT's existing True 3D Holographic Angular Slice 3D (HAS3D) FoLD displays is shown in Figure 6. The workstation concept would be an extremely attractive platform for testing the SMFoLD standard in a typical command center environment.

TDT's FoLD technology has a number of advantages for command centers:

(a) No glasses required for 3D viewing;
(b) Parallax "look around" viewing--move one's head to see around objects;
(c) Reproduces all human visual cues, long term viewing without eye fatigue;
(d) Continuously blended perspectives, no pseudoscopic "flipping" or dead zones;
(e) Excellent brightness, no need to dim room lights or close shades for viewing;
(f) Compatible with existing 3D apps (QT Modeler, Google Earth, AGI's STK, GXP);
(g) No head or eye tracking, avoids motion lag or other tracking issues;
(h) No moving parts such as spinning discs, excellent solid state reliability;
(i) Scalable design for display sizes from desktops to conference rooms to theaters.

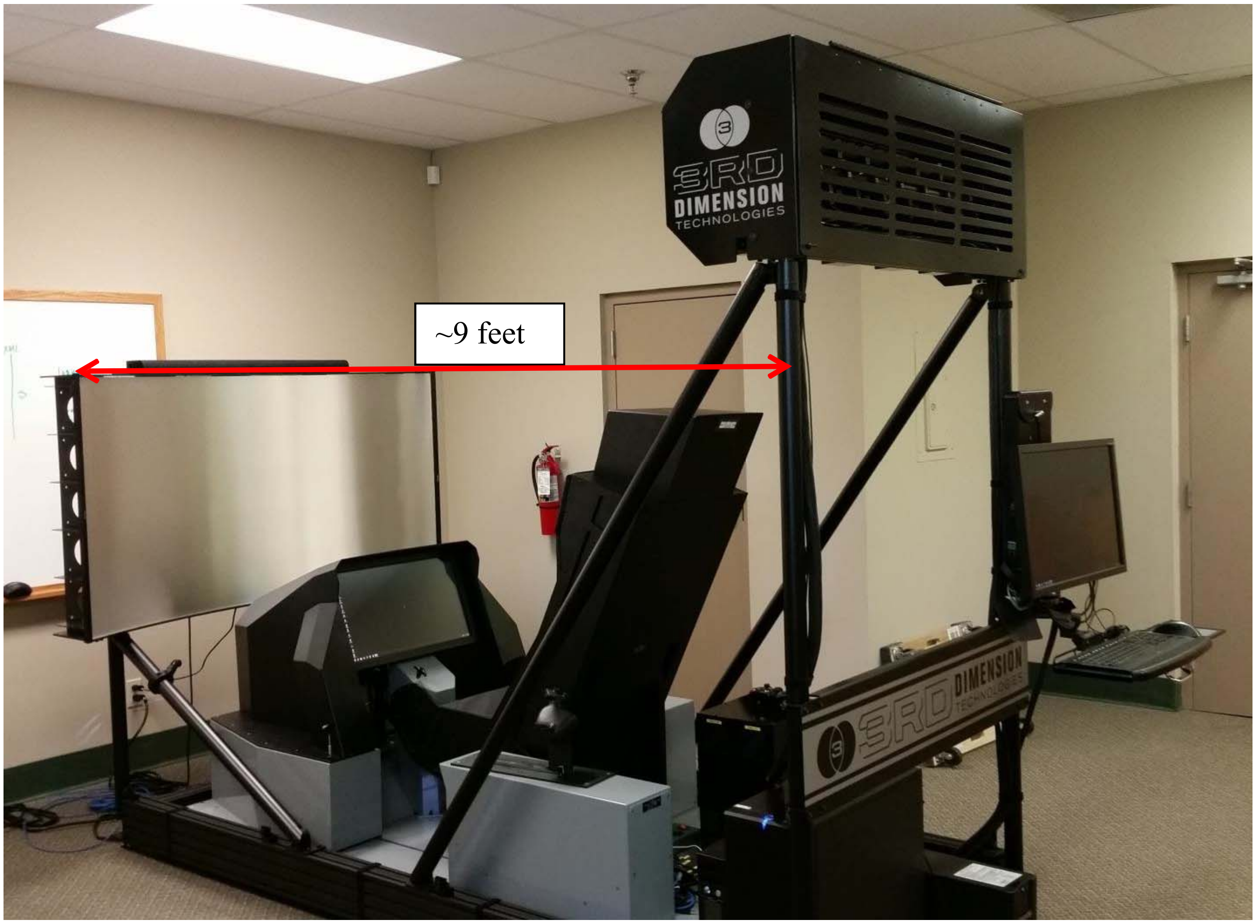

Figure 5. Image of Flight Simulator with Integrated TDT High Resolution FoLD System

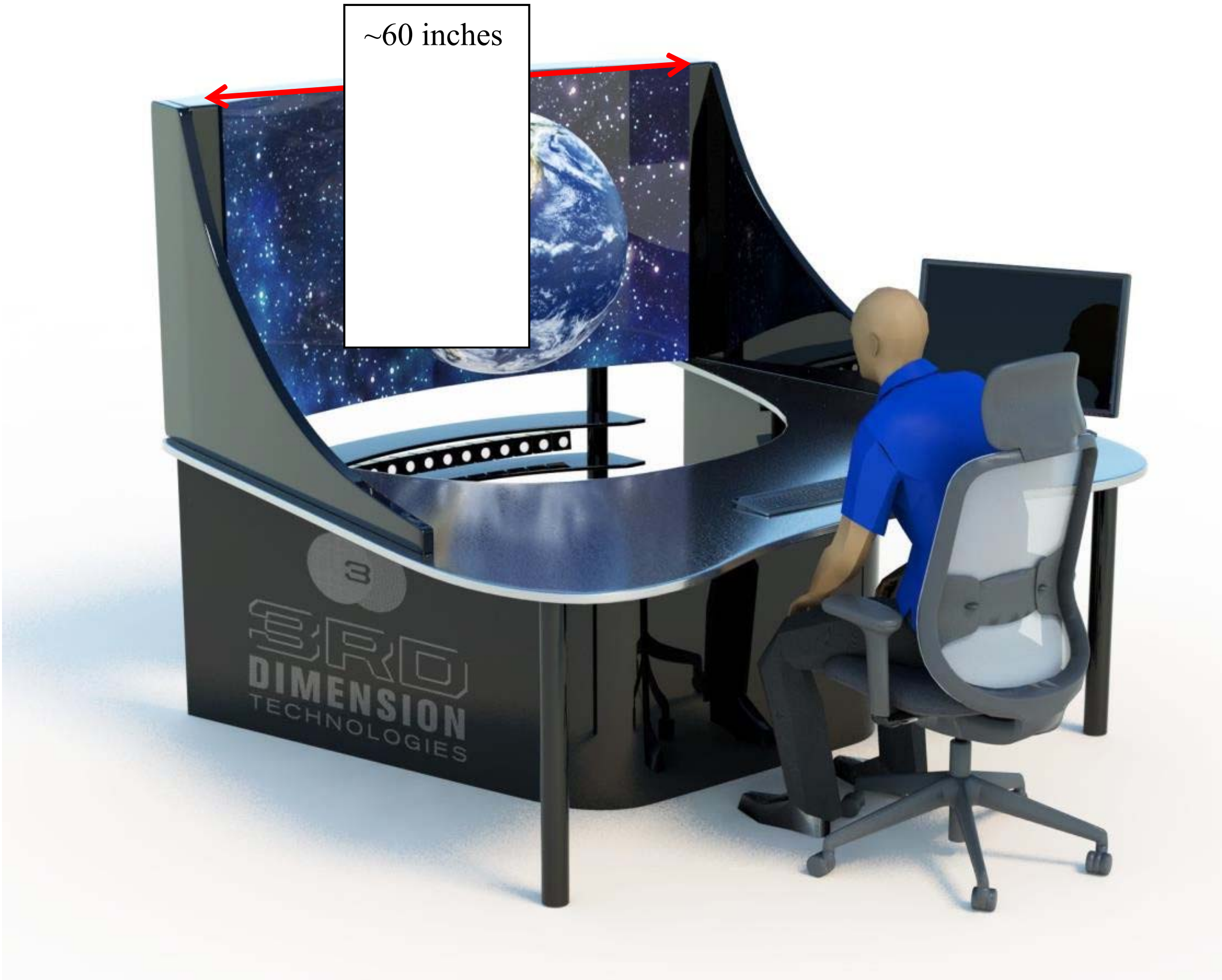

Figure 6. Conceptual Design of High-Resolution Horizontal Parallax SMFoLD Compliant Workstation

3.0 METHODS, ASSUMPTIONS, AND PROCEDURES

The Air Force has identified a need for the creation of a common streaming model for 3D data—including a scene description protocol and transmission format—that is agnostic to the display technology at the end user. To address this need, TDT and ORNL have completed work on an STTR Phase II project to develop an open standard for display agnostic 3D streaming of Field of Light Display data (FoLD data). The standards effort is in general referred to as SMFoLD.

The TDT team has successfully completed the research and development tasks for Phase II and has implemented SMFoLD. Work on Phase II of the SMFoLD project commenced on 19 September 2017 and was completed 18 December 2019. SMFoLD workshops were held on 3 October, 2017 and 2 October 2018. An SMFoLD website was created in Phase I and updated during Phase II to facilitate communication among interested parties. SMFoLD was briefed at technical conferences including SD&A and Display Summit and in a webcast hosted by SMPTE. Papers on SMFoLD were published in the proceedings of SD&A and a paper has been submitted to the SMPTE Motion Imaging Journal.

Deliverables for the Phase II contract and their status are shown in Table 3 below.

Table 3. Contract Deliverables Status

Deliverable	Contract Item	Due Date	Status
Status Report #1	0001AA	18 Oct 2017	Complete
Status Report #2	0001AB	18 Nov 2017	Complete
Status Report #3	0001AC	18 Dec 2017	Complete
Status Report #4	0001AD	18 Jan 2018	Complete
Status Report #5	0001AE	18 Feb 2018	Complete
Status Report #6	0001AF	18 Mar 2018	Complete
Status Report #7	0001AG	18 Apr 2018	Complete
Status Report #8	0001AH	18 May 2018	Complete
Status Report #9	0001AJ	18 Jun 2018	Complete
Status Report #10	0001AK	18 Jul 2018	Complete
Status Report #11	0001AL	18 Aug 2018	Complete
Status Report #12	0001AM	18 Sep 2018	Complete
Status Report #13	0001AN	18 Oct 2018	Complete
Status Report #14	0001AP	18 Nov 2018	Complete

Status Report #15	0001A1	18 Dec 2018	Complete
Status Report #16	0001AR	18 Jan 2019	Complete
Status Report #17	0001AS	18 Feb 2019	Complete
Status Report #18	0001AT	18 Mar 2019	Complete
Status Report #19	0001AU	18 Apr 2019	Complete
Status Report #20	0001AV	18 May 2019	Complete
Status Report #21	0001AW	18 Jun 2019	Complete
Status Report #22	0001AX	18 Jul 2019	Complete
Status Report #23	0001AY	18 Aug 2019	Complete
Status Report #24	0001AZ	18 Sep 2019	Complete
Charts for Review Meetings #1 - #12	CDRL A003	31 Oct 2019	Complete
Nonproprietary Phase II Summary Reports	CDRL A004	19 Sep 2019	Complete
Software User Manual	CDRL A005	18 Dec 2019	Complete
Software Test Plan #1	CDRL A006	18 Jun 2018	Complete
Software Test Plan #2	CDRL A006	18 Jun 2019	Complete
Software Test Report #1	CDRL A007	18 Sep 2018	Complete
Software Test Report #2	CDRL A007	18 Dec 2019	Complete
Software Source Code and Executables	CDRL A008	18 Dec 2019	Complete
Draft Data Streaming Model for Field of Light Display (FoLD) Visualization Systems #1	SOW Task 1.0	17 Dec 2017	Complete
Draft Data Streaming Model for Field of Light Display (FoLD) Visualization Systems #2	SOW Task 6.0	18 Dec 2019	Complete
Draft Final Report	0001BA	18 Oct 2019	Complete
Final Scientific and Technical Report	0001BB	18 Dec 2019	This Report

3.1 SMFoLD Website

The SMFoLD website (http://www.SMFoLD.org) was created and published September 4, 2016. The site was used to announce the SMFoLD workshops with pages for registration, agenda, call for speakers, and links to presentations and videos. The blog page and initial post were added September 8, 2016 and the Wiki was added October 3, 2016. Presentation and videos from the first workshop (held Friday, 28 Oct 2016) were posted to the website. Presentation abstracts were generated and are available at http://www.smfold.org/agenda/ (Scroll down beneath the agenda to see the abstracts). Presentations and videos are available on the same page. The Wiki and blog are located at http://www.smfold.org/wiki and http://www.smfold.org/blog respectively. The SMFoLD website banner is shown in Figure 7.

The Announcement (http://www.smfold.org/2017-smfold-workshop/) and Agenda (http://www.smfold.org/2017-smfold-workshop/) for the 3 Oct 2017 SMFoLD Workshop were added in late September 2017, and the presentations were posted to the website shortly after the Workshop was completed (http://www.smfold.org/2017-smfold-workshop/smfold-2017-presentations/). Short abstracts, headshots, and bios can be found by clicking on the presentation title on the Agenda page.

The agenda and abstracts for the 2 Oct 2018 SMFoLD Workshop can be found at http://www.smfold.org/2018-smfold-workshop/.

The SD&A paper and presentation from the Electronic Imaging 2018 conference are posted on the SMFoLD website Blog.

Figure 7. SMFoLD Website Banner

3.2 SMFoLD Workshops

Two SMFoLD Phase II workshops were held, one on 3 Oct 2017 and another on 2 Oct 2018. These workshops are discussed in some detail in APPENDIX D.

3.3 AFRL FoLD Workshops

FoLD workshops was held at Wright-Patterson AFP in November 2017, 2018, and 2019 by the AFRL 711th HPW/RHCV, Dr. Darrel Hopper. The first day was dedicated to identifying Government needs and non-proprietary presentations by 3D display industry leaders and researchers. Day two was dedicated to giving researchers and vendors an opportunity to present the state of their research to Government representatives without non-government attendees present. Proprietary information could be revealed on day two. The FoLD workshops are discussed in APPENDIX E.

4.0 RESULTS AND DISCUSSION

Based on the research and efforts of TDT and ORNL, and on the information gathered and reported on from the SMFoLD (see below) and FoLD workshops in the previous section, this section discusses an actual SMFoLD implementation.

4.1 Introduction and Background

4.1.1 Government and Industry Needs

Command and control battle maps and mission planning can be greatly enhanced by the representation of the battlefield in three dimensions (3D). Showing perspective views, depth and occlusions allows for a better understanding of the situation, and assists in avoiding errors when interpreting the terrain. Many sources of 3D data such as Digital Terrain Elevation Data (DTED) maps already exist and are in use, but are projected using two-dimensional display devices which limit their effectiveness.

Also of interest is the ability to make sense of large data sets, including the ability to view large numbers of objects and have an understanding of their spatial relationships. This understanding applies to a wide variety of applications from medical imaging to the visualization of space junk. Avoiding errors in analyzing these data can be critical.

4.1.2 Implementation Challenges

It has been shown through the numerous presentations that producing display agnostic 3D visualization using pixel data is computationally and transmission prohibitive. Streaming 3D geometric data with all of the information a 3D display requires reduces the bandwidth for streaming but at the cost of photo realistic scene production. Adding texture data and high quality materials properties can help achieve the required level of realism. The display is ultimately responsible for producing the pixel data from the geometric data. While this approach requires the generation and transmission of many layers of sometimes complex data to reconstruct images that rival their video captured counterparts, it is ultimately a more flexible approach allowing layers to be added or resolution to be increased as the user desires and the bandwidth permits.

SMFoLD implementation is conceptually simple, using one 3D data type (Open GL mesh and texture data and other OpenGL graphics primitives), based on a proven paradigm that TDT uses on its True3D HAS3D systems. Other 3D data types (plenoptic data, point cloud data, CAD data, …) can optionally be supported in the future. Note that point cloud data is supported by OpenGL. Streaming of complete 3D data frames produces a protocol similar to existing 2D protocols with both key frames (complete 3D data frames) and change frames (changes to Vertex Buffer Objects and Vertex Array Objects already stored on the Display System GPU or memory).

4.1.3 Current TDT HAS3D FoLD Implementation

TDT has many years of experience with light field displays and has built on that expertise. The current method used to display 3D data on TDT's True 3D Holographic Angular Slice Display (HAS3D) system is in several ways analogous to streaming except that the data flows over a hardware bus in a computer instead of over a network. TDT's TitaniumGL (TiGL) software intercepts OpenGL function calls that an application makes to the GPU and re-renders multiple images from the different corresponding image viewpoints and positions to create a true 3D image with horizontal parallax.

A well-known technique is employed to intercept the OpenGL commands, or functions, used by an existing application, e.g. Google Earth. Since the commands are intercepted the interceptor is free to perform any operations deemed necessary. In the HAS3D system, TiGL encodes the OpenGL commands as numerical values, or opcodes, and the length of the associated data, the opcode, and the associated data are written to a memory buffer. Some functions require special handling and need to do more than the actions described above. Opcodes are transmitted rather than OpenGL function calls in order to compress the data.

When the 3D frame is complete the entire command sequence with associated data is in a memory buffer that is shared by the rendering processes for all the viewpoints, one rendering process for each viewpoint. These rendering processes nominally all run in parallel. An event is fired that signals the rendering processes that the scene is ready to be processed. The rendering processes read the memory buffer and convert the opcodes and data into OpenGL function calls and call the real OpenGL library functions. The result is that the rendering processes replicate all of the actions of the source application with a custom viewpoint for each HAS3D projector.

With this HAS3D TiGL implementation, viewers get a different view to each eye with true horizontal parallax. Each view has a Gaussian overlap with the next view, and these overlaps are computer-blended together so that a continuous scene with no dead spots or inversions is observed by the viewer. TDT has built on this HAS3D TiGL paradigm to implement the SMFoLD standard for both HP and full parallax (FP) systems.

4.1.3.1 Parallelized Multi-View Rendering

While not proposed as part of the SMFoLD standard, TDT has parallelized the rendering of viewpoints so that all the GPUs are rendered in parallel. A process is defined for each GPU, with all the viewpoint information (camera positions, camera angles, FOV) for that GPUs cameras in the Process memory, passed down from the Display Application. Each viewpoint parses the equivalent of the SMFoLD 3D Frame (a shared memory buffer for HAS3D) and loads its own geometry data variables for the TDT HAS3D implementation.

In general all SMFoLD compliant Display Applications will be able to parallelize their rendering process applications (to the extent allowed by the graphics card API) for each viewpoint to provide FOV, data extent, POV and other metadata (to be determined). This allows the rendering pipeline for each viewpoint to run in parallel with all the other viewpoints (again, to the extent allowed by the GPU API), so that there is no need to serialize the viewpoint rendering across

GPUs. In general GPUs can presently be parallelized, but different views a on a single GPU run serially. This is true for both OpenGL and Microsoft's DirectX11 API.

All of this will also be made possible for an SMFoLD compliant Display Application because the SMFoLD standard will allow the Source Application to provide metadata via the View matrix. Each viewpoint server process can set up its own geometry and run its rendering pipeline in parallel with the other GPUs, though separate viewpoints on a single GPU are presently serialized.

4.1.4 Advantages of the Proposed SMFoLD Standard

The proposed SMFoLD standard has several advantages over other potential solutions:

4.1.4.1 The data size of an SMFoLD 3D Frame is very small compared to the amount of plenoptic data required to create multiple viewpoints.

4.1.4.2 The data size of an SMFoLD 3D frame is very small compared to sending multiple compressed video streams for each viewpoint.

4.1.4.3 By allowing metadata for each viewpoint via the OpenGL View matrix, the SMFoLD standard is agnostic to the hardware and software geometry of any particular FoLD system.

4.1.4.4 Any platform that currently uses OpenGL 3.3 or above to render 3D scenes can be easily adapted to support the new SMFoLD standard. The following are required for SMFoLD Source Application compatibility:

(a) Link to an SMFoLD library

(b) SMFOLD.dll file placed in the application directory

4.1.4.5 Any Display Application can easily become SMFoLD Compliant by implementing the following:

(a) The SMFoLD Library translation of Opcodes to OpenGL function calls must be made available to the Display Application by linking to the SMFoLD Display DLL.

(b) A reference implementation (not part of the SMFoLD standard) will be made available as source code for Display Applications

(c) Note that Display Applications are not required to use OpenGL. Given the Opcode translation to OpenGL the Display Application can choose to use any API or graphics engine that it chooses to implement the functionality.

4.1.4.6 Basing the SMFoLD standard on the well-known OpenGL standard will facilitate the development of new SMFoLD compliant applications.

4.1.4.7 The popularity of OpenGL will increase the likelihood that the SMFoLD standard will be widely adopted.

4.1.4.8 The approach is inherently flexible so that 2D, 3D, FoLD or tiled displays can be supported.

4.2 Create SMFoLD Standard Technical Report

A draft SMFoLD Standard Technical Report (TR) has been created and delivered to AFRL on 18 December 2017. The draft report discusses the actual implementation of the FoLD Streaming Model to transport 3D frames to multiple types of displays, including FoLD displays. The revised SMFoLD Standard Technical Report was delivered to AFRL in December 2019.

4.3 Implement SMFoLD Standard

4.3.1 Overview

4.3.1.1 Introduction

Current methods for streaming content to monitors, televisions, virtual reality (VR), Stereo 3D (S3D), and Augmented Reality (AR) devices are insufficient for Field of Light Displays. FoLD systems require rendering the same geometry from many viewpoints to create parallax. A display with N degrees of horizontal parallax requires N viewpoints to be rendered at partial degree offsets. A full parallax display with both horizontal and vertical parallax requires N^2 (or N x M if asymmetric) viewpoints to be rendered. The bandwidth required to stream raster data to a full parallax display at video rates will quickly exceed commercially available internet connection speeds since it grows by a factor of N^2. The SMFoLD approach is to stream OpenGL function calls and render all the views at the display.

In the SMFoLD approach illustrated by a block diagram in Figure 8, a Source Application renders the central view of a scene and a Display Application running on a FoLD system executes multiple passes of the render loop to generate the offset views. The SMFoLD Source DLL intercepts the API calls made by any OpenGL application that links the SMFoLD Source DLL Library. The function calls are serialized, compressed, and transmitted securely to a remote machine running a Display Application which renders 1 to N views as required by the target display type. Since the offset views share the same texture and geometry data, only one copy needs to be transmitted to the remote machine.

The current SMFoLD implementation is based on the Modern OpenGL 3.3 core profile. The SMFoLD interface (SMFoLD Source DLL) is an OpenGL shim (a shim is a piece of software that intercepts API calls from an application program and manages the API calls in some fashion, such as substituting an alternative call, or passing them on to another application) that transparently intercepts API calls. All OpenGL function calls made by a Source Application are intercepted and forwarded to both the system OpenGL DLL for local display and the SMFoLD DLL Packer for further processing.

Since function pointers on the local system will not be valid on a remote system, an integer opcode is assigned to each OpenGL API call. The Packer shown in Figure 8 serializes each API call as a chunk of binary data containing the opcode and function parameter data. The binary data is then compressed by an encoder (codec) and inserted into a thread safe, concurrent queue to await streaming to the remote machine.

OpenGL calls are pushed into the concurrent queue by the encoder and popped from the queue by the WebSocket client/server. A concurrent queue allows both respective threads to operate on the queue safely.

The WebSocket server negotiates a Transport Layer Security (TLS) 1.3 connection with a client. New public/private encryption key pairs are generated for each session. Only recommended cipher suites and hash functions specified in the Federal Information Processing Standard (FIPS)

Publication 140-2 Annex A are utilized. Once a connection is established the server begins dequeuing the serialized and compressed OpenGL function calls and transmitting them over the established WebSocket secure (WSS) connection.

The Websocket client runs on the remote computer and connects to the WebSocket server on the source computer via a WebSocket secure (WSS) connection. The data received is inserted into a thread safe, concurrent queue to await further processing by the unpacker and decoder. The remote OpenGL function queue will exactly mirror the calls made locally on the source computer.

The decoder decompresses the data received by the WebSocket client. The Unpacker then deserializes each chunk of binary back into an opcode and function parameter data. The opcode is replaced with the proper function pointer for each OpenGL API call on the Display platform.

OpenGL function calls are ready to be executed in the remote system's render loop after being processed by the unpacker and decoder. A Display Application that is tailored to the target display type manages the render loop. In the case of a 2D monitor, the Display Application will render one view that essentially duplicates the view rendered on the source computer. A horizontal parallax FoLD system will render N viewpoints by replaying the streamed OpenGL calls and changing the camera parameters N-1 times to generate N offset views. A full parallax FoLD system will follow the same process in order to generate N^2 (or N x M) views offset both horizontally and vertically.

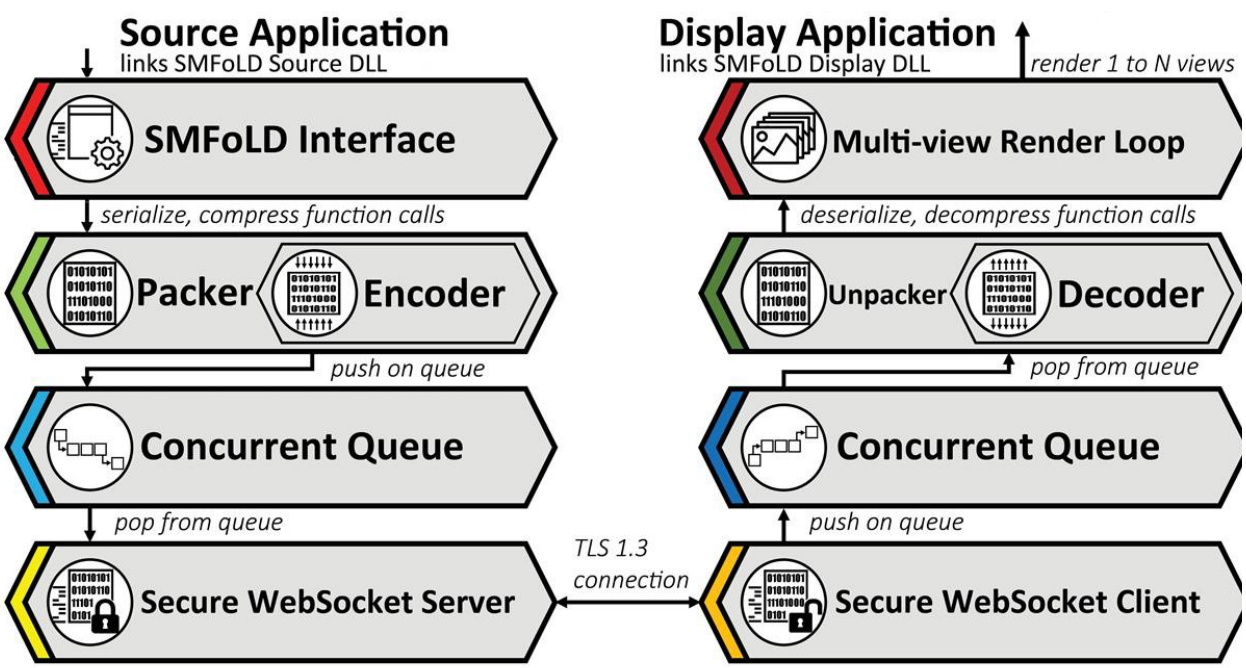

Figure 8. Block Diagram of SMFoLD Workflow

4.3.1.2 SMFoLD Source Library (DLL) Design

The default logical behavior of SMFoLD.dll has been designed such that buffer packing threads don't block when waiting for a wsClient connection. This allows the Source Application to render frames and execute normally while waiting for a client to connect. Though the Test Source Application was not negatively impacted by a forced pause, it stands to reason that this could cause problems in more complex programs.

The buffer is still packed every frame but is only sent when a client is connected. Subsequent client connections after a disconnection resync with the current Source Application frame. This behavior mirrors that of popular live streaming services such as Amazon's Twitch.tv and Google's YouTube live streams. Frames generated prior to initial client connection or during the time elapsed between reconnections are dropped. This is the most logical default behavior; however, additional modes could be implemented where a client could request a buffer of specified duration be maintained on disconnect. In this scenario no large contiguous chunks of frames would be dropped, but the client stream would not be live until some subsampling of the replay frames allows the client to catch up to the current Source Application buffer.

4.3.1.3 SMFoLD Display Application Design

The Display Application now enforces a limit of one SMFoLD Server process per machine. This change was made in order to force compliance with programming best practices for parallel OpenGL rendering presented at the AMD Fusion Developers Summit.[63] A single SMFoLD Server process instantiates multiple SMFoLD Server class objects to generate the desired number of rendered OpenGL views. The suggested configuration is one SMFoLD Server class object per GPU where each object controls an OpenGL viewport with the desired number of views.

4.3.1.4 SMFoLD Display Application Hardware Considerations

The number of views per GPU is only limited by the complexity of the scene relative to the speed of the hardware. The optimal configuration will of course be application dependent but the following example can serve as guide to the thought process. As an example, a Radeon Pro WX 9100 is capable of rendering current generation games (a computationally demanding task) at 3840 x 2160 (4k) resolution at a sustained 60 frames per second on average. This is roughly equivalent to rendering four 1920x1080 (1080p) views at 60 fps. Since there is presently a practical limit of four GPUs per computer, a system with a sufficiently fast CPU could be expected to render sixteen 1080p views at 60 fps. This configuration would use a Display Application with four SMFoLD Server class objects, each rendering a quad view 1080p OpenGL viewport.

Currently available COTS projectors that meet the physical size requirements for use in the HAS3D system have a native resolution of 1280x800 (WXGA). Since the Radeon Pro WX 9100 has six video outputs the practical limit on the number of rendered views is twenty-four given four WX9100 GPUs. Though this is significantly less computationally demanding than the 1080p example, the refresh rate limitation of the projectors would limit the displayable frame rate to 60 fps. This configuration would use a Display Application with four SMFoLD Server class objects, each rendering a six view WXGA (1280 x 800 pixels) OpenGL viewport.

A dual CPU socket system could raise the practical limit on GPUs to eight. Intel's Skylake-SP and AMD's Epyc product lines both contain CPUs with sufficient Peripheral Component Interconnect express (PCIe) lane configurations to run eight cards. This configuration would use a Display Application with eight SMFoLD Server class objects, and could potentially render up to thirty-two 1080p views or forty-eight WXGA views.

SMFoLD's design is scalable within the realm of current possible configurations that could be built with COTS hardware, though driver limitations could prevent the theoretical maximums from being realized,

4.3.1.5 Multi-GPU Rendering

In order to demonstrate the speedup that can be achieved on a Display Application by rendering with multiple GPUs (for the case of slow rendering at the display) ORNL did an experimental measurement of the rendering speedup that might be achieved with two GPUs vs. one GPU, for a case relevant to the LightSpace x1406C display.

Multi-GPU rendering or parallel rendering is a technique to achieve near linear scaling with the number of GPUs to improve primitive throughput, fill rate or to handle larger datasets. It consists of parallelizing geometry processing (transformation, clipping, lighting, etc.) and rasterization (scan-conversion, shading, visibility determination, etc.). Parallel rendering consists of three steps:

(a) Distribution of the workload/rendering tasks to all GPU's
(b) Collection of rendering results from all GPUs
(c) Composition of the rendering results into the final image

According to the distribution of workload, parallel rendering can be classified in sort-first, sort-middle and sort-last.[64]

Sort-first distributes primitives early in the rendering pipeline, i.e. during geometry processing. This is generally done by dividing the screen into disjointed regions. Each region is then assigned to a processor that in turn will do all the calculations of its respective region. When primitives fall in screen regions other than the one on which they reside, they must be redistributed to the appropriate processor.

In sort-middle, geometry processing and rasterization are performed by separate processors and primitives are redistributed between geometry processing and rasterization. Arbitrary subsets of primitives are assigned to geometry processors and portions of the screen are assigned to rasterizers. During each frame, geometry processors transform, clip, light, etc. their portion of the primitives and classify them with respect to screen regions. Then, the geometry processors transmit all the screen-space primitives to the appropriate rasterizer for final rendering.

The sort-last method assigns arbitrary subsets of the primitives to each processor. Each processor then executes geometry processing and rasterization independently no matter where the primitives fall on the screen. Processors then transmit these pixels to "compositing" processors to resolve the visibility of pixels and display.

In recent years, sort-first and sort-last techniques have been used extensively to support ultra-high resolution framebuffers and large scale datasets respectively. As mentioned before, communication overhead is involved in both techniques due to primitive redistribution and final rendering compositing. In a multi-GPU setting, intermediate results are shared among GPUs through the PCIe bus and using the CPU as a data broker which limits scalability. To alleviate this limitation, AMD and NVIDIA have designed two ad-hoc solutions: AMD's Crossfire and NVIDIA's SLI.

4.3.1.5.1 Light Field Parallel Rendering

Light field rendering outputs rasterized primitives to fixed-size framebuffers with vertical or horizontally shifted projection views. The number of framebuffers and its corresponding projection views depends on the FoLD system configuration. In the case of the LightSpace x1406C display, the system renders twenty different views with a resolution of 1024 x 768 pixels each.

The ORNL approach to FoLD parallel rendering is to distribute rendering of each view across several GPUs. While the sort-last method is traditionally used to support large scale datasets (i.e. when datasets does not fit in one GPU memory) being displayed in a fixed-size window, ORNL adapted this method for parallel rendering. Modifications are listed next:

(a) Instead of assigning arbitrary subsets of primitives to each GPU, the primitives are replicated across GPUs:

(b) Rendering occurs independently across GPUs; rendering parameters remain the same except for the projection view.

(c) Each GPU is in charge of displaying a subset of views independently, thus sort-last's visibility check for compositing is not needed, but a synchronization step is required to avoid flickering.

4.3.1.5.2 Implementation

Supporting multi-GPU rendering is dependent on the parallel rendering algorithm, OpenGL's context creation (e.g. WGL, GLX), operating system, and the vendor solution (e.g. AMD Crossfire, NVIDIA SLI) used. ORNL implemented a FoLD parallel rendering (FoLD-PR) prototype following the steps described in the previous section. ORNL also enumerated potential modifications that need to be done to the SMFoLD client:

(a) FoLD-PR prototype uses Message Passing Interface (MPI)[65] for multi-processing, synchronization and data interchange. In contrast to traditional multi-threaded processing, MPI enables FoLD-PR prototype to create processes and manage communications at inter and intra node level.

(b) For simplicity and to avoid OpenGL specifics in multi-threading settings, an MPI process is created per rendering view and a copy of the primitives exists on each process. Rendering parameters remain the same except for the projection view.

(c) FoLD-PR prototype follows a master-worker scheme, i.e. the master process controls User Interface (UI) event registration and tracking (mouse and keyboard) using the OpenGL Utility Toolkit (GLUT) and broadcasts updates to worker nodes. Since GLUT can only create the OpenGL context on GPU-0, the master node always uses GPU-0 for rendering. Workers receive UI events and update their views accordingly.

(d) FoLD-PR prototype implements custom OpenGL context creation code using OpenGL Extension to the X Window System (GLX) under Linux to allow worker processes to access additional GPUs. A similar approach can be adopted on Windows operating system using OpenGL Extensions for Microsoft Windows (WGL).

(e) Although FoLD-PR prototype is based on the sort-last algorithm, in FoLD-PR gathering rasterization output from each process is not needed for final compositing, thus ad-hoc solutions such as AMD Crossfire or NVIDIA SLI are not required. The only data that is shared across MPI processes is the UI's interaction events and synchronization flags.

(f) The source code of the FoLD-PR prototype is based on a previously published technique for in-situ visualization of crowd simulations in GPU clusters.[66] The source code for FOLD-PR is located on TDT's SMFoLD GitHub site and will be made publicly available when approved by AFRL.

4.3.1.5.3 Test Results

In order to verify the performance of the FoLD-PR prototype, a test was designed to render a variable number of instances of a character mesh positioned at the center of a virtual environment. The light field was made of twenty viewports with a resolution of 1024 x 768 pixels and the character mesh was made of 2,788 vertices or 4,963 triangles.

The configuration of the system used in this test included an Intel Haswell CPU i7-4770S running at 3.10GHz with four cores and eight threads, 32 GB of RAM, and two NVIDIA Turing Quadro RTX 5000 GPUs.

First, the FoLD-PR prototype was run on one GPU. The algorithm performance (time per frame) is reported in Figure 9 and Table 4. Note that the number of triangles equals number of instances x number of triangles per instance x number of viewports.

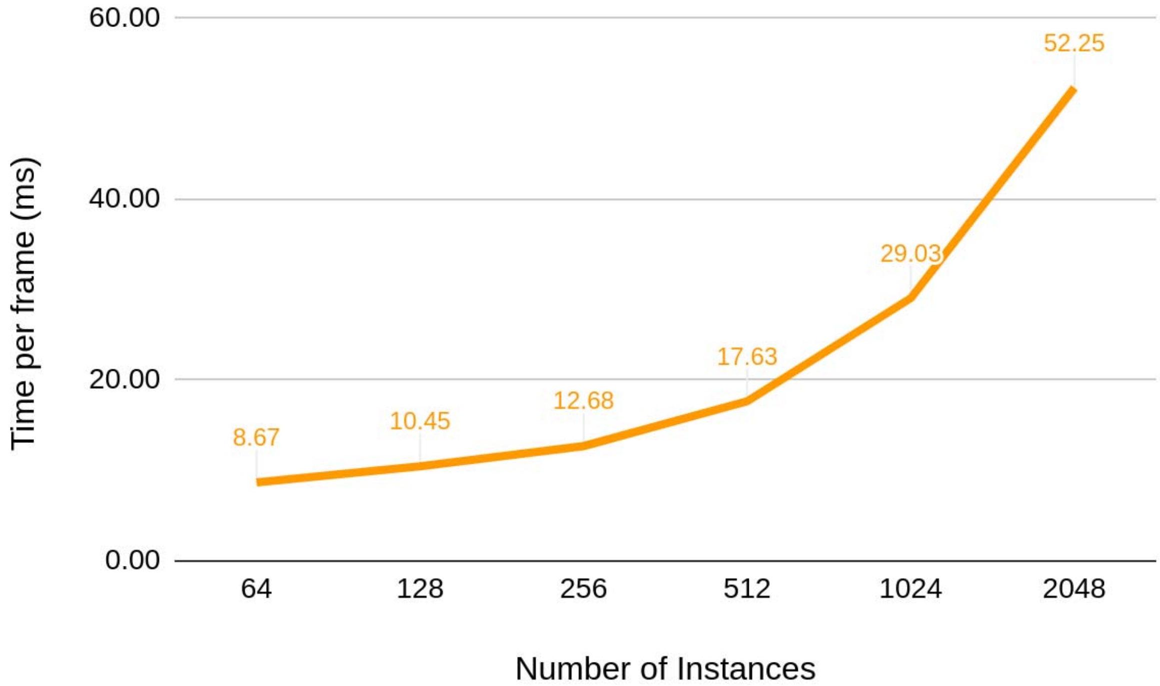

Figure 9. Performance Results of the FoLD-PR Prototype Using One GPU

Table 4. Performance Results of FoLD-PR using One GPU

# Instances	# Triangles	Time per frame (ms)
64	6,352,640	8.67
128	12,705,280	10.45
256	25,410,560	12.68
512	50,821,120	17.63
1024	101,642,240	29.03
2048	203,284,480	52.25

Figure 10 and Table 5 report performance results when using two GPUs. In this case, the workload was divided in such a way that each GPU rendered ten viewports.

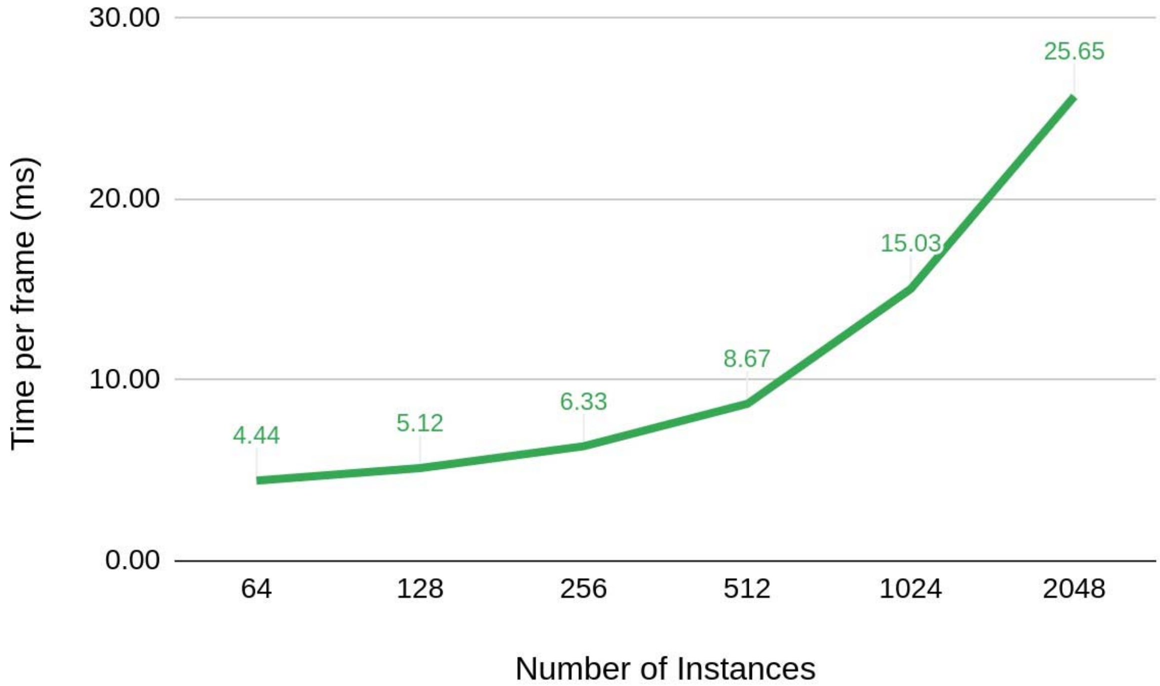

Figure 10. Performance Results of the FoLD-PR Prototype Using Two GPUs

Table 5. Performance Results of FoLD-PR using Two GPUs

# Instances	# Triangles	Time per frame (ms)
64	6,352,640	4.44
128	12,705,280	5.12
256	25,410,560	6.33
512	50,821,120	8.67
1024	101,642,240	15.03
2048	203,284,480	25.65

Speed up obtained when using two GPUs is reported in Figure 11 and Table 6. Overall, FoLD-PR prototype rendering time was around 1.99 times faster on a two GPU configuration vs one GPU, which indicates the solution scales linearly when adding an additional GPU. Also notice the fact that the 4-core/8-thread CPU was oversubscribed by using twenty MPI processes, however performance was not affected since the CPU was only used to communicate the UI events among processes, leaving the main rendering tasks to the GPUs.

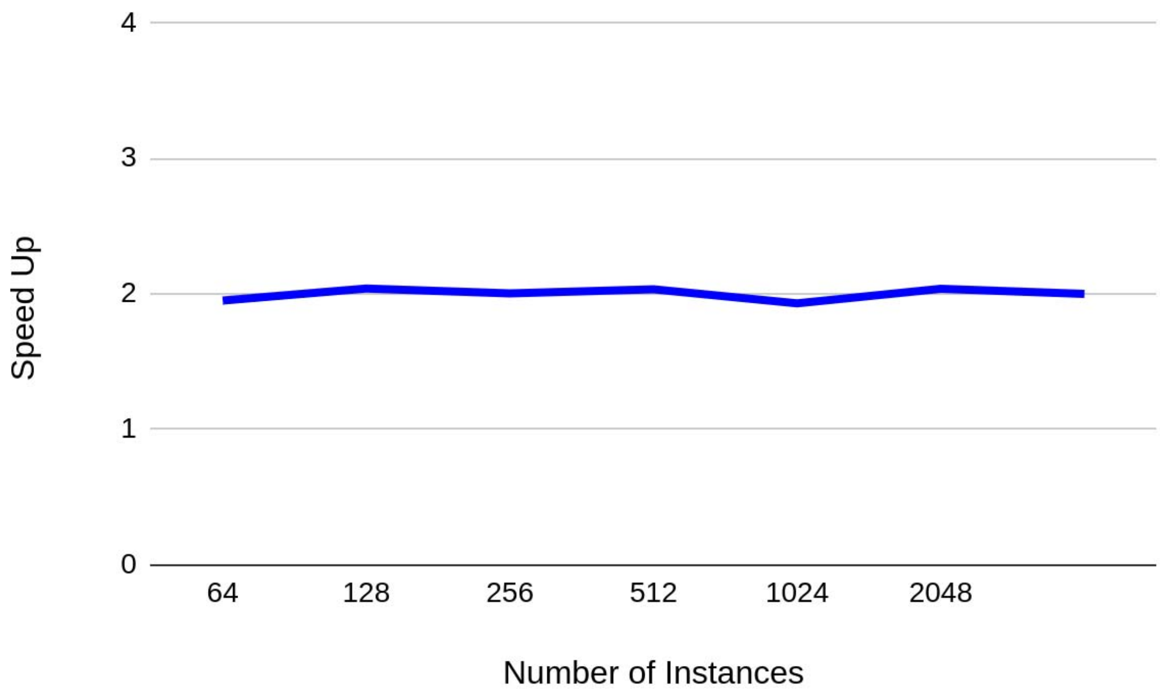

Figure 11. FoLD-PR Speed Up When Using Two GPUs

Table 6. FoLD-PR Speed Up When using Two GPUs

# Instances	Speed Up
64	1.951148131
128	2.039828192
256	2.002842255
512	2.033210332
1024	1.931403859
2048	2.037037037

4.3.2 Develop SMFoLD Source and Display Process

The first Modern OpenGL 3.3 release of SMFoLD was completed on 15 Dec 2018. This SMFoLD release uses OpenGL Core 3.3 functions. SMFoLD currently demonstrates end to end operation using a simple demo Source Application with a limited command set. This release will compile and run without requiring any additional libraries to be installed or environmental path variables to be set.

4.3.2.1 SMFoLD Modern OpenGL

The current SMFoLD version uses OpenGL 3.3 core, though support for versions 4.0 through 4.5 and beyond could be added in the future without significant changes to the SMFoLD architecture. Version 3.3 core was selected because it is the first OpenGL version to force programs to use Vertex Array Objects (VAOs) and/or Vertex Buffer Objects (VBOs) for loading 3D geometry. Using VAOs and/or VBOs allows the GPU to only copy geometry into memory once and manipulate the scene through state changes in lieu of additional CPU to GPU memory synchronizations. This is inherently more efficient and can lead to considerable performance gains over Legacy OpenGL's immediate mode.

Supporting OpenGL 3.3+ core also guarantees that any Source Applications will be able to implement capabilities such as change frames that are defined in the SMFoLD standard. In fact, all 64bit OpenGL 3.3+ core applications will implicitly be SMFoLD compliant. Note that Source Applications running older versions of OpenGL can still be SMFoLD compliant if they follow good design practices and use the programmable pipeline as well as VBOs and/or VAOs rather than fixed function pipeline and/or immediate mode.

Ideally the SMFoLD Source must implement each and every function call defined in the target graphics API. At the present time only a limited subset of OpenGL 3.3 core function calls have been implemented. The first step in planning for the implementation process was to generate a function call list. An offline version of the Modern OpenGL API Reference was generated and saved to a local disk. The online version can be viewed at the link below for reference:

https://www.khronos.org/registry/OpenGL-Refpages/gl4//

Using regular expressions and string processing, list files containing the function call prototypes and version compatibility information for API version 2.0 through 4.5 were generated by parsing the reference page html files. The results showed that the OpenGL 3.3 core specification contains three-hundred and forty unique API calls. The function prototypes utilize eight distinct return types and forty-five parameter data types that in combination form one hundred and ninety unique function signatures with frequency counts shown in the Figure 12 graph below.

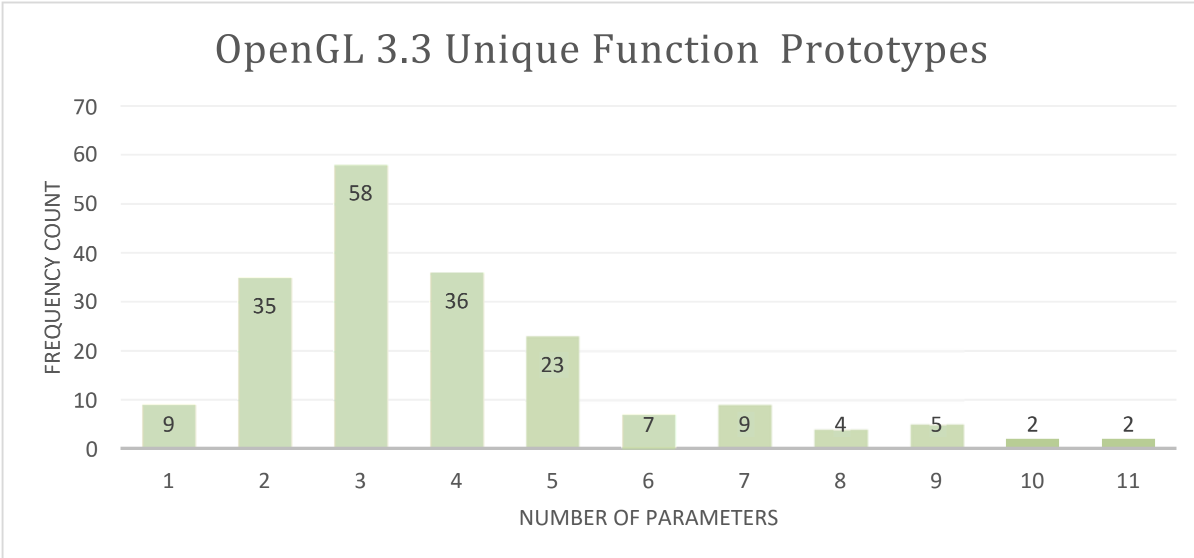

Figure 12. OpenGL 3.3 Unique Function Counts Sorted by Parameter Count

Build scripts have been created to generate the OpenGL interceptor code for each function as well as the packing and unpacking parameters used in SMFoLD Source. Knowledge of the size that each data type will occupy in memory will allow a fairly generic script to generate code for each unique function prototype in each frequency bin. Functions with pointers instead of data as parameters need special casing since SMFoLD must transmit the referenced data over WebSockets rather than the pointer itself.

As shown in Table 7 there are 336 core functions to be implemented for OpenGL 3.3 compatibility. Those that can be implemented with a script have been completed. Advancing to OpenGL 3.3 has been a major escalation of the original project goal to deliver SMFoLD as OpenGL 2.1 compliant. OpenGL 3.3 and follow-on releases are often called "Modern OpenGL" since they completely eliminate fixed pipeline rendering and create the rendering pipeline with programmable shaders (GLSL—GL shader language). Also as shown in Table 7, one of the important features of OpenGL 3.3 core is that it is compatible with and supported by all following versions of OpenGL, up to and including OpenGL 4.6. Note also that change frame capability is implemented in OpenGL by forcing the use of Vertex Buffer Objects (VBOs) and Vertex Array Objects (VAOs), prior to OpenGL 3.2 this was optional. VBOs and VAOs must be used and manipulated by the Source Application program, but they are stored by the Display Application program either in memory or on the GPU card. Implementation of change frames and methods to optimize compression and frame latency are further discussed below.

Table 7. Function Count for OpenGL Versions 2.0 to 4.5

OpenGL Version	Function Count by Version	Functions Added Since Previous Version	Cumulative Function Count Since 2.0 Core	Deprecated Functions Supported
2.0 Core	191	-	-	Y
2.1 Core	197	6	6	Y
3.0 Core	281	84	90	Y*
3.1 Core	293	12	102	N
3.2 Core	331	38	140	N**
3.3 Core	336	5	145	N**
4.0 Core	363	27	172	N**
4.1 Core	433	70	242	N**
4.2 Core	445	12	254	N**
4.3 Core	488	43	297	N**
4.4 Core	497	9	306	N**
4.5 Core	601	104	410	N**

* legacy functions marked deprecated but not removed from core profile
** legacy functions removed from core, supported with compatibility profile

The high-level concepts and roles associated with each component of SMFoLD remain mostly the same in both the legacy and modern OpenGL versions. However, the modern OpenGL version introduces considerable changes and improvements in the underlying software implementation of the architecture. Most notably, the multi process shared memory model has been changed to a more modern multi-threaded object-oriented design. This change necessitated refactoring the names of some project components. Source Process and Display Process libraries are now named SMFoLD Source and SMFoLD Display respectively. The "process" designation no longer makes sense; under the multi-thread model the only distinct processes are Source Application and Display Application. An updated block diagram that reflects the current state of the components of SMFoLD is shown in Figure 13 below:

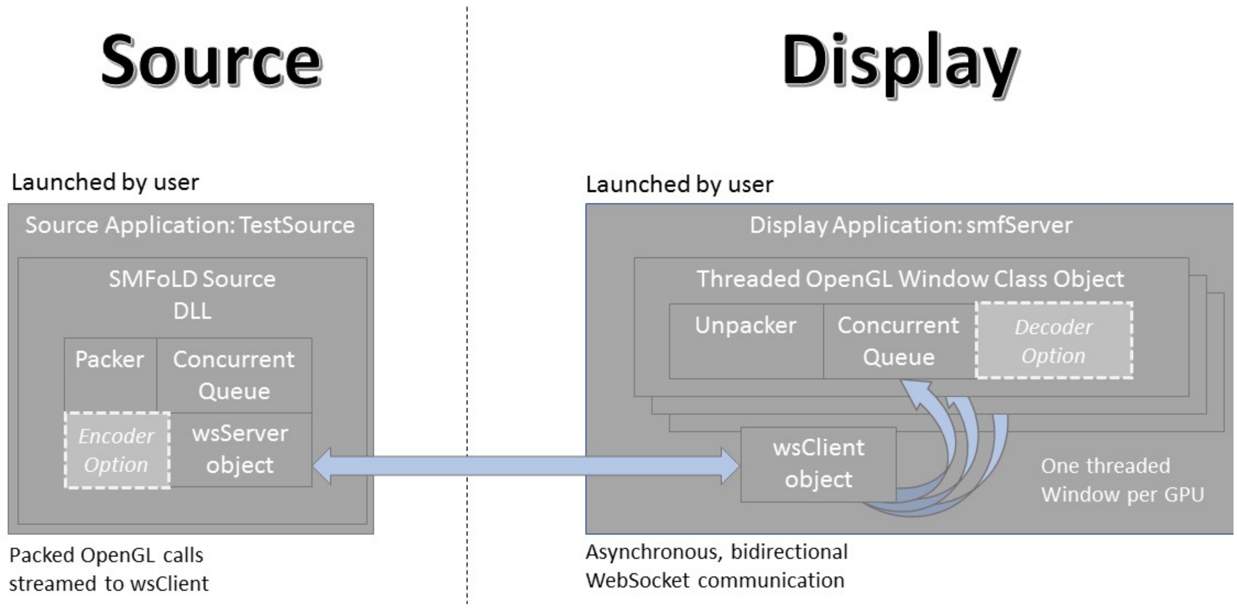

Figure 13. Simplified Block Diagram of the Interactions Between SMFoLD Compliant Source and Display Applications

During the process of upgrading support from legacy OpenGL 2.1 up to modern OpenGL 3.3, all legacy C project code was also rewritten and/or replaced with C++ code. The C++ standard template library (STL) as well as production quality third party libraries such as Intel's Thread Building Blocks and Boost provide easy access to a variety of data structures and algorithms. The switch to C++ allowed development to focus on high level design rather than low level implementation when solving problems. For example, the development of data serialization needed to transmit OpenGL calls over WebSocket connections used well documented, robust, open source solutions written in C++. SMFoLD uses the MessagePack library for data serialization. The main benefits of MessagePack are code simplicity, data type safety, and protection from buffer overruns.

4.3.2.2 OpenGL Call Interception

For OpenGL 3.3 functions that have been implemented, SMFoLD implements a function with the same prototype as each function defined in the OpenGL standard. SMFoLD.dll exports the SMFoLD function name as the name of the OpenGL function with the same prototype. This causes OpenGL function calls in an application that links SMFoLD.dll to call an SMFoLD function instead of the expected OpenGL function. The resulting interceptor then forwards calls to the system OpenGL32.dll and packs and serializes the function pointer and parameter data for streaming. Pseudocode for a generalized case of the interceptor for functions with no return type is shown below:

```
SMFoLD_Function(type0 param0, type1 param1, … , typeN paramN)
{
glFunction(param0, param1, … , paramN);
packFunction(&API_Call_Queue, param0, param1, … , paramN);
}
```

The function parameters are packed in a Standard Template Library (STL) tuple and then serialized as a binary string of unsigned characters. Parameters for the generic function example above would be packed as an N element tuple. Since the addresses of the function pointers will differ between a remote machine and the host machine, each is serialized as a short integer. Each packed OpenGL function call can be represented with a short integer opcode and a binary string of serialized parameter data. Pseudocode for this packing process is shown below:

```
glFunction(type0 param0, type1 param1, … , typeN paramN)
```

params are packed as:

```
tuple<type0, type1, … , typeN>
```

param tuples are serialized and then calls are then packed as:

```
tuple<short,string>
```

The storage container for the packed OpenGL calls is now a Thread Building Blocks (TBB) concurrent queue rather than a shared memory buffer. Thread Building Blocks is Intel's open source C++ template library for parallel programming. Using a concurrent data structure fixes the intermittent buffer synchronization problem experienced in previous implementations. The other benefit is that dynamic memory allocation makes the amount of system memory the only limitation on frame size. No a priori knowledge of scene complexity or frame size is required.

4.3.2.3 Initial SMFoLD Implementation

The OpenGL 3.3 core call interceptor and packer prototypes have been implemented for all functions that have void or data type parameters. These were able to be automatically generated by parsing the OpenGL specification documents. The remaining portion is comprised of functions that have a pointer parameter. These pointer functions require hand coding and consultation of the OpenGL specification documents; a slow and tedious process. The SMFoLD initial release contained only the pointer type functions implemented that are required for the Test Source Application. The initial release consisted of the following project files.

SMFoLD Project File Summary:

SplashScreen

- SplashScreen: Displays a splash screen window with an image background and buttons to "Start SMFoLD" or "Quit". There is a placeholder space that can be used to display a user notice or liability clause.

Source Application

- TestSource: A simple OpenGL SMFoLD compliant source applications that displays a colored cube in a borderless window. The cube position responds to mouse movement inside the drawable area which provides a qualitative gauge of latency. Other Source Applications will be available.

SMFoLD_Source (formerly Source Process, now a DLL)

- packer: responsible for intercepting OpenGL calls and packing each function call's corresponding opcode with its parameter data into a concurrent API call queue that serves as a frame buffer.
- wsServer: WebSocket client that transmits the contents of the concurrent API call queue to a wsClient object or process.
- SMFoLD: combines wsServer, packer, and the export interface into SMFoLD.dll which is linked by SMFoLD compliant Source Applications.

SMFoLD_Display (formerly Display Process, now a DLL)

- wsClient: Currently only used for testing purposes in combination with wsServer. smfServer negotiates WebSocket connections using an instance of the wsClient class object rather than a separate process.

Display Application

- smfServer: Manages a wsClient instance and a concurrent API call queue. Packed API calls are unpacked and executed as OpenGL function calls. The demo program demonstrates rendering four different views (quad viewport) of the colored cube in the Test Source Application.

4.3.2.4 SMFoLD Initial Test Results

The SMFoLD initial release was tested on the hardware listed below. The only known hardware requirements are a 64-Bit processor with 4 or more cores (4 cores/8 threads minimum recommended) and an OpenGL 3.3+ compatible video card. The tested hardware list is provided to establish a few sample points across a gamut of high-end professional down to entry level gaming hardware.

OS:
Windows 10 Professional 64-Bit

GPU:
Radeon Pro WX 9100
Nvidia GeForce GTX 1080
Nvidia GeForce GTX 1060

CPU:
Intel Core i9-7900X Skylake-X
Intel Core i9-8950HK
Intel i7-3770K

A flickering artifact in the Display Application occurred when pairing a high-end GPU with a high refresh rate monitor (120hz+) due to frames being rendered faster than the SMFoLD 3D frame buffer could be filled. This has been addressed in a later update with a form of framerate flow control and improvements to the buffer fill rate.

TestSource.exe and smfServer.exe must have adequate permission settings to run and access the network interface. Anti-Virus and Firewall software that perform auto containment and/or virtualization sand-boxing are very likely to prevent proper program execution. Sand boxing also prevents the application from knowing that system calls and I/O operations failed, which can prevent proper error messages from being displayed. Firewall settings can also stop the program from running.

An intermittent condition where shaders failed to link correctly was experienced during testing. This could have been due to either the Source Application or Display Application loading a window then quickly exiting. No action is needed if this error occurs as the next run of the program usually succeeds. This failure condition has been addressed.

The Display Application, smfServer.exe, does not run on Windows 7. The Source Application can be run on Windows 7 64-Bit as long as the Display Application is run on a Windows 10 machine and connected across the network.

A screen shot in Figure 14 below shows the debug output of a successful run.

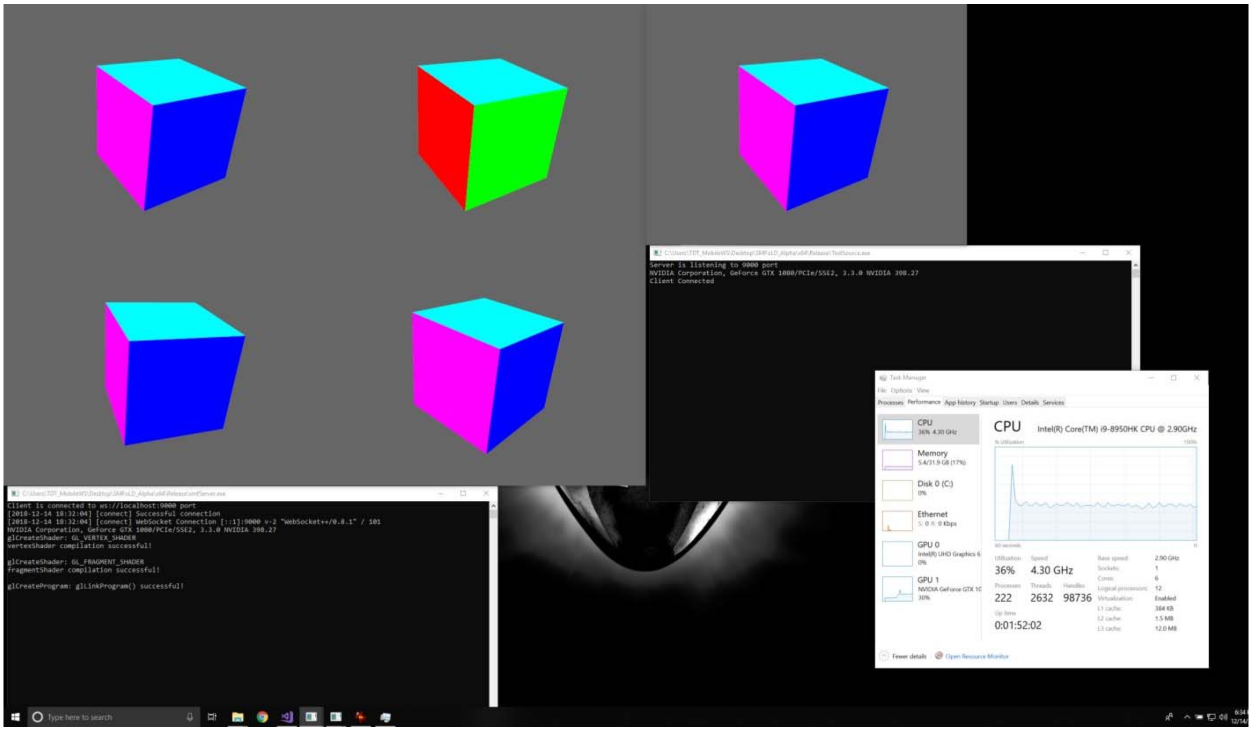

Figure 14. Debug Output of a Successful Run of the SMFoLD Initial Release

4.3.3 Platform for Demonstration and Test (TDT/ORNL)

SMFoLD Example Calibration Software work has been completed to provide easy to use and flexible calibration software for FoLD systems. The calibration software will not be a part of the SMFoLD standard but will be a demonstration of how to implement Display Application calibration that is compatible with the SMFoLD software and demo Display Application The software currently provides the Demo Display Application with projector location, projector orientation and polynomial-based spatial transformation information in an XML format. The SMFoLD Demo Display Application consumes this information to provide the unique camera model for each of the projectors.

TDT is currently using the calibration software output files as input for the SMFoLD Demo Display Application on TDT's 22 projector systems. The calibration software is compatible with any N, or N x M projector FoLD system.

Initial use of the LightSpace x1406C volumetric display in Figure 15 has been demonstrated on TDT hardware using a DirectX StarWars example provided by LightSpace. As discussed below, SMFoLD has been successfully implemented on the x1406C volumetric display.

Figure 15. LightSpace Volumetric Display

4.3.3.1 LightSpace Display Application SMFoLD Implementation

The LightSpace x1406c volumetric display is comprised of 20 depth planes of 1024 x 768 pixels each. The video signals are routed from a single 4k resolution frame buffer. The sample Display Application TDT developed for the LightSpace x1406c addresses the display with a single OpenGL window containing 20 viewports in a one window, one GPU configuration. Since the x1406c is capped at 20 Hz due to DisplayPort 1.2 bandwidth limitations, there is generally no practical value in using more than one GPU for rendering. The NVIDIA Quadro and Quadro RTX GPUs that are recommended by LightSpace are capable of well above 20 fps at 4k even with complex content. In the event that multi GPU rendering was needed, multiple SMFoLD borderless windows could be displayed side-by-side such that each GPU is responsible for an equal number of columns or rows. Render loop logic would have to be modified to maintain the proper depth slice order, but no other substantive changes would be required. The depth plane order for the one GPU, one window configuration is shown in Figure 16 below.

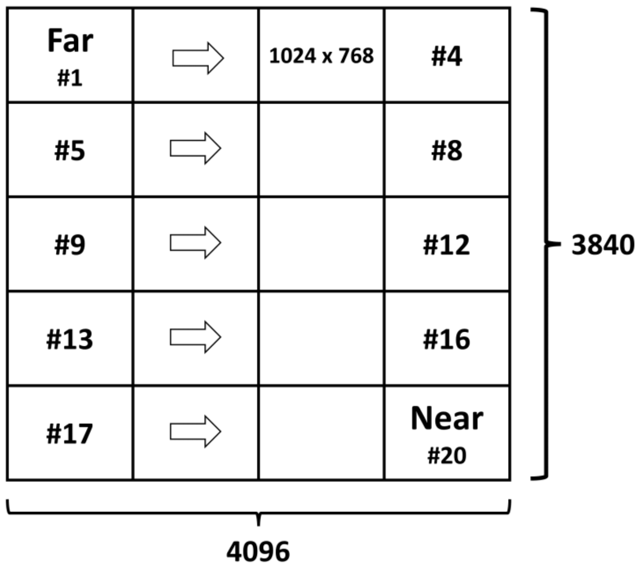

Figure 16. Depth Plane Order and Configuration for LightSpace x1406c OpenGL Window Utilizing one GPU

A LightSpace Display Application depth slice vertex shader implementation was accomplished in order to properly render a scene on the LightSpace x1406c volumetric display, the desired scene must be sliced into 20 segments spaced evenly along the camera line of sight between the near and far clipping planes. These intermediate clipping planes are parallel to the near and far planes and orthogonal to the camera line of sight vector. The equation below is an easy way to construct a plane.

$$Ax + By + Cz + D = 0$$

In terms of computer graphics, vector <A,B,C> is a surface normal to a clipping plane at depth D. In brief, multiplying a point by the View Projection matrix maps the point from eye space to Normalized Device Coordinates (NDC). Likewise, a point in NDC space can be mapped to eye space by multiplying by the inverse of the View Projection matrix. From this relationship, it follows that the front face of the NDC cube maps to the camera frustum's near plane and the back face of the NDC cube maps to the far plane via multiplication by the inverse View Projection matrix. Likewise, a vector along the z-axis in NDC space ((0,0,-1) to (0,0,1)) then maps to the camera's line of sight in eye space. The view frustum and NDC cube are depicted in Figure 17 below.

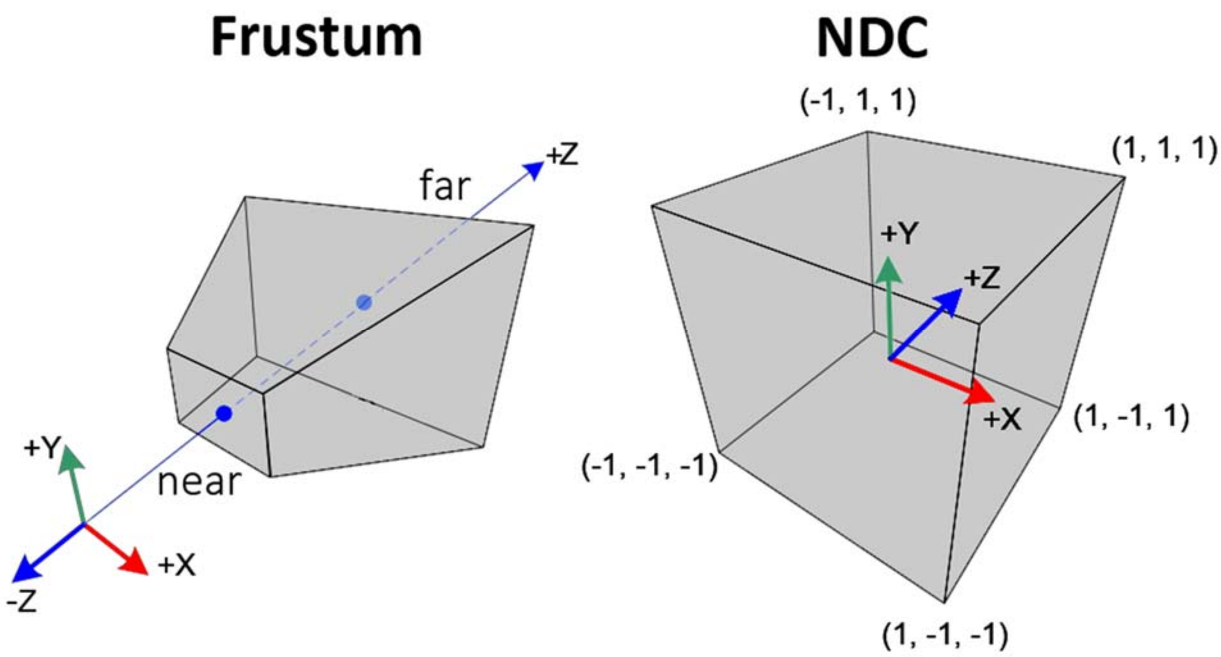

Figure 17. Camera View Frustum and NDC Cube

The relationships described above lead to the following algorithm which is implemented as a vertex shader:

(a) Calculate the normalized (unit) line of sight vector for the current View Projection matrix in eye space. This the vector between points (0,0,-1) and (0,0,1) in NDC space. Multiply each by the inverse View Projection matrix to get these points in eye space. Then, construct the vector and normalize it.

(b) Define the extents of the scene to be rendered. This will generally be the near and far plane for the current View matrix, but could be intermediate values as well.

(c) Divide the extents from step 2 into 20 clipping volumes. Each volume is defined by two clipping planes such that the surface normal is the vector calculated in step 1 and the depth values are the intermediate near and far plane depths for each of the 20 volumes.

(d) Use the set of planes constructed in step 3 to define custom clipping planes for each viewport in the vertex shader. The sign of the dot product of a vertex and each clipping plane indicates if the vertex is in front or behind the plane. Reject all vertices in front of the near plane or behind the far plane. Creating a slight overlap between slices and alpha blending can lessen the gaps between slices that are visible when the view is not centered on the display screen.

The majority of the technical effort for driving the x1406C has been focused on developing the Display Application and associated shaders for the LightSpace x1406C volumetric display. The LightSpace Display Application is complete though there are additional refinements to the shaders that could improve image quality such as overlapping the depth slices and alpha blending the leading and trailing edges. A Display Application for the LaunchTN 22-channel HP has been completed and demonstrated on 5 Dec 2019.

4.3.3.2 SMFoLD Integration on 22-Channel LaunchTN Demo System

A picture of the projector layout for the 22-Channel System is shown in Figure 18.

Figure 18. Projector Array From 22-Channel System

Integration of SMFoLD on the 22-channel demo system uses AMD Eyefinity Display Groups. Each display group is comprised of all the projectors connected to a respective GPU. In the case of the 22-channel system, there are four GPUS and four display groups. Figure 19 shows the Display Group mapping to the projectors.

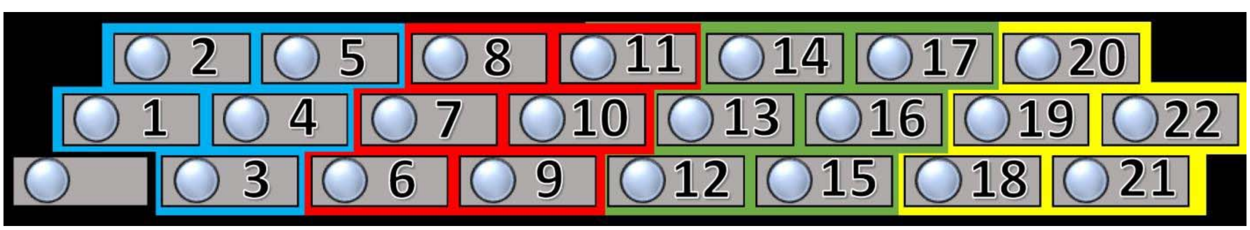

Figure 19. Display Groups Matched to Projectors in Array

Each display group is associated with one SMFoLD Display Application window instance. An SMFoLD window fully spans each display group. OpenGL viewports segment the window to align with the projector pixel space layout. There are unique XML config files implemented for each viewport. The XML config files provide a one-to-one correspondence between projector numbers (physical label) and viewports. Viewports represent the projector layout on the Windows Desktop as shown in Figure 20.

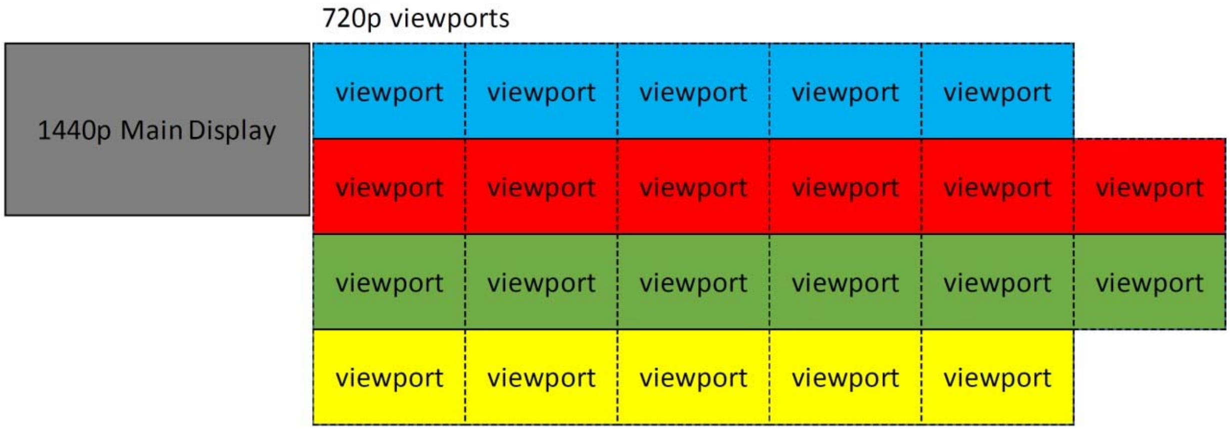

Figure 20. Viewport Layout in Windows Desktop

As mentioned above work has been completed to implement and demonstrate SMFoLD on the LaunchTN system.

4.3.4 Change Frame Capability

Storing Change Frames -- the geometry data in a vertex buffered object (VBO) or vertex array object (VAO) -- allows state changes such as object translation, rotation, scaling, material properties, and lighting without making superfluous copies of the geometry data. The geometry data only needs to be transmitted once and can either remain in the remote graphics card's memory or be cached on the remote machine. Reducing geometry data transfers can dramatically reduce the memory and networking bandwidth required to send SMFoLD frame data at high frame rates. For example, each floating-point vertex in an uncompressed SMFoLD frame requires 14 bytes in memory. A ten million vertex model would then require 140 MB of memory uncompressed. Sending state updates in lieu of complete, redundant geometry can reduce the required space in memory by multiple orders of magnitude. In this example, the 140 MB frame would be reduced to hundreds of bytes (a reduction on the order of 104). Since Modern OpenGL versions 3.3+ require the use of VBOs or VAOs for geometry data storage, all SMFoLD compliant applications will natively support and require this OpenGL programing paradigm.

4.3.5 Compression Capability

The SMFoLD team has developed a codec (compression/decompression and encryption/decryption) to provide secure and high performance transmission of data using the SMFoLD standard (see Figure 21).

Figure 21. SMFoLD Frame Transmission

4.3.5.1 Codec Benchmarks

ORNL has explored two algorithms for compression, Blosc[106] and Zstandard[107]. Both offer two characteristics relevant for the SMFoLD protocol: fast compression speed at reasonable compression rates. Blosc is designed to transmit data to the processor cache faster than the traditional, non-compressed, direct memory fetch approach via a memcpy() OS (memory copy operating system call) call. Blosc is the first compressor that reduces the size of large datasets on-disk or in-memory and accelerates memory-bound computations.[106]

Blosc uses the blocking technique to reduce activity on the memory bus as much as possible, i.e. it divides datasets in blocks that are small enough to fit in L1 cache of modern processors and performs compression/decompression there. It also provides SIMD (SSE2) (Single Instruction Multiple Data, Streaming SIMD Extensions 2) and multi-threading capabilities so as to accelerate the compression/decompression process.

Blosc works as a compressor engine that accelerates traditional compression/decompression algorithms supporting LZ4 and LZ4HC, Snappy, zlib and Zstd algorithms (all of these are compression toolkits/algorithms). This supported diversity is useful to test different speed/compression rate/algorithm trade-offs.

The Zstandard compression algorithm extends the zlib algorithm to offer faster compression speed (up to 5x), better compression ratio (10-15 percent smaller) and 2x faster decompression speed. Zstandard was designed to be a good approach in many different cases, i.e. when the data need to be processed many times, decompression speed and the ability to opt into a very high compression ratio without compromising decompression speed is advantageous. Zstandard is also good at small data[107], in particular in JSON (Java Script Object Notation) messages between a web server and a browser.

Blosc and Zstandard provide implementation in C/C++[108,109] with Linux compatibility. Deployment of such libraries in Visual Studio is not as straightforward as it is in Linux. ORNL has completed work on generating libraries to provide support to the SMFoLD Visual Studio base.

Encryption is based on the Version 1.3 of the Transport Layer Security (TLS) Protocol standard as proposed under RFC 5246[110] (Request for Comments: an Internet Engineering Task Force (IETF) document) by the Network Working Group. This protocol allows the SMFoLD Source DLL and Display DLL to communicate in a way that is designed to prevent eavesdropping, tampering, and message forgery.

The encryption protocol is composed of two layers – the TLS Record Protocol and the TLS Handshake Protocol. The Record Protocol ensures that (i) the connection between the SMFoLD Source and SMFoLD Display is private and (ii) the connection is reliable. The Handshake Protocol ensures that the client's identity can be authenticated using asymmetric cryptography and that the negotiation of the shared secret is secure. Additionally this protocol establishes reliability of the negotiation – a man-in-the-middle modification would be detected by both the client and server.

Symmetric cryptography is used for the data encryption. The keys for the symmetric encryption are generated uniquely for each connection. Two open source TLS encryption implementations were investigated: cryptlib[111] and S2n.[112]

In Table 8 below, the performance of various compression/decompression algorithms is displayed. These algorithms were tested on an Intel i7-7700K CPU using four threads (deployed by Intel in Q1, of 2017, runs at 4.2 GHz in standard mode).

Table 8. Compression Performance of Various Algorithms

Compressor Name	Size In *(bytes)*	Size Out *(bytes)*	Compression Factor	Compression Time *(ms)*	Decompression Time *(ms)*
blosclz	41,943,040	383,449	109.4	3.7	3.2
lz4	41,943,040	539,770	77.7	3.4	3.5
lz4hc	41,943,040	381,308	110	6.2	3.4
snappy	41,943,040	2,091,963	20	3.1	4.5
zlib	41,943,040	285,328	147	38.9	15
zstd	41,943,040	158,428	264.7	53.5	4.3
blosclz	83,886,080	745,273	112.6	5.2	6.7
lz4	83,886,080	901,306	93.1	4	7.2
lz4hc	83,886,080	742,844	112.9	11.2	7
snappy	83,886,080	4,185,083	20	6.1	7.8
zlib	83,886,080	505,487	166	74.5	24.6
zstd	83,886,080	289,112	290.2	101.1	7.2
blosclz	125,829,120	1,083,193	116.2	6.3	10.2
lz4	125,829,120	1,238,266	101.6	5.7	10.4
lz4hc	125,829,120	1,079,804	116.5	16.8	10.6
snappy	125,829,120	6,277,723	20	8.9	12.9
zlib	125,829,120	737,247	170.7	11.5	38.8
zstd	125,829,120	422,872	297.6	146.9	10.8

As can be seen from the table results, and as discussed above, Blosc uses techniques to speed up compression that make it very attractive for the SMFoLD codecs. See Table 9 below for possible frame rates with varying levels of compression.

Table 9. Possible Frame Rates with High Speed Internet

Example Application	Frame Rate (FPS) At % Compression (C) Network Speed 1Gbps (125MB/s)						
	Raw 3D Frame Data Size (Bytes)	C	FPS	C	FPS	C	FPS
		25%		50%		75%	
Google Earth	2,049,837	1,537,377	81.3	1,024,919	122	512,459	244
Poles	1,248,427	936,320	133.5	624,214	200.3	312,107	400.5
3D Fish	4,279,805	3,209,853	39	2,139,902	58.4	1,069,951	116.8
QT Reader	6,705,819	5,029,364	24.9	3,352,909	37.3	1,676,454	74.6

4.3.5.2 Latency and the Memory Wall

Over the past few decades the number of instructions per second executed by a processor and the capacity per memory chip has consistently doubled every two years. In contrast to this trend, memory latency has decreased at a much slower rate. The growing differential between the improvements in compute performance and memory latency leads to a condition commonly referred to as the “Memory Wall.” The implication being that maximal compute performance cannot be realized if data streams are unable to feed the processor fast enough. Avoiding the Memory Wall and optimizing latency reduction are key factors in ensuring that remote streaming performance closely approximates the experience and expectations of running 3D applications on a local machine. The main computational components and latency sources for SMFoLD are depicted in Figure 22 below.

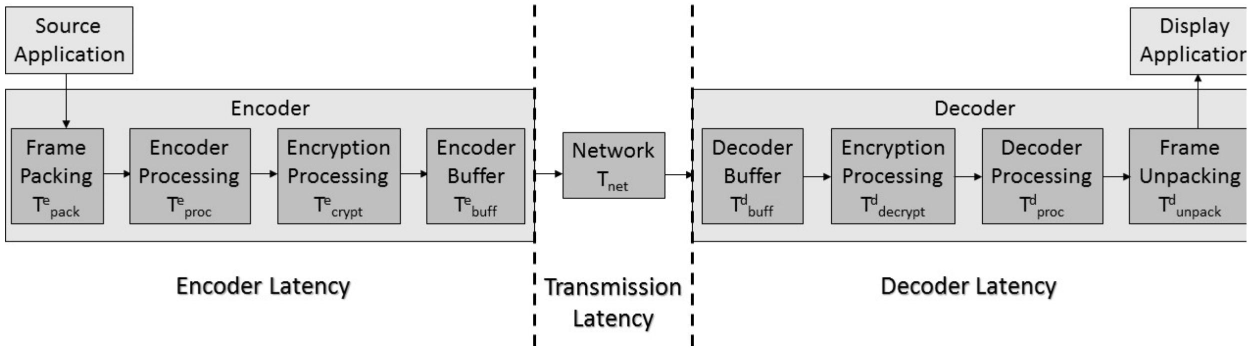

Figure 22. Block Diagram of Compute Steps that Generate Latency in SMFoLD Pipeline

Considerable resources are spent in the High-Performance Computing (HPC) market to advance the state of the art. Many lessons learned from analyzing HPC networking paradigms, data processing software design patterns, and Memory Wall mitigation strategies are directly applicable to SMFoLD.

Compressing SMFoLD frame data can both reduce the networking transmission bandwidth requirements and potentially mitigate the memory wall. Blosc, a blocking, shuffling and lossless compression library, can in fact outperform C++ memcpy when used with a low compression ratio. Blosc could be used in this manner when copying frame data into buffers for further processing such as compression and/or encryption. Blosc with a high compression ratio will serve as a codec that will be applied just prior to encryption.

Reducing the frame data size reduces transmission latency. However, the latency introduced by encoding must be carefully weighed against the reduction in transmission latency for optimal total latency. Maximizing compression ratio could be counter-productive if it takes too long. The optimal values will balance minimizing both encoding and transmission latency.

Another approach to mitigating the Memory Wall is to reduce the number of buffer copies performed. Remote Direct Memory Access (RDMA) allows a network adapter in a workstation or server to transfer data directly into application memory on a remote system. This zero-copy networking allows data transfers into a remote system's memory without burdening the sender or receiver CPU. It also saves both systems an additional memory buffer copy that would normally be required to transfer the data from the network adapter into system memory.

RDMA capability is available with many commercial off the shelf (COTS) network adapters. Since support starts with 10GbE it is important to note that it is COTS hardware, not commodity hardware. RDMA over Converged Ethernet (RoCE) is an example network protocol that is common on HPC and enterprise hardware that leverages RDMA. RDMA capable network adapters could significantly improve SMFoLD performance when available as two less buffer copies will occur per frame. RDMA will be recommended but not required by the SMFoLD standard since compatible hardware is uncommon outside HPC and enterprise channels.

4.3.5.3 Other Considerations

Blosc and many other compression algorithms work best when the data being operated on can be stored in CPU cache instead of in memory. Many 3D applications will generate SMFoLD frames that are larger than the available CPU cache. Even the simple poles demo application generates frames that are larger than most CPU L1 cache at 640kb per frame. As an example, Intel's flagship i9 Skylake-X CPUs have 256kb of L1 data cache per core.

In order to efficiently handle larger frames, data will need to be streamed as cache sized or smaller sub blocks rather than a single contiguous frame data block. In this case frame boundaries will be demarcated by frame buffer swap API calls rather than the data buffer extents. OpenGL draws API calls sequentially to an off-screen back buffer and then swaps the front and back buffer to display the drawn frame. This is a significant change from the previous SMFoLD Source Process design, but will produce identical functionality. Sending frames in small fixed sized chunks rather than as complete frames improves compression efficiency and reduces latency. Switching to streaming small fixed size data segments also makes the SMFoLD transport model conceptually similar to the approaches used by large streaming media companies such as Netflix, Amazon's Twitch.tv, and Apple.

4.3.5.4 Codec Implementation

The SMFoLD Codec Status is as follows. A Blosc based codec has successfully been implemented by ORNL and integrated into SMFoLD by TDT. After packing and serializing OpenGL function data from the Source Application, compression is applied and the compressed data is temporarily stored and pushed into a concurrent queue. The data is then popped from the queue and transmitted over a WebSocket connection in the SMFoLD Source DLL's Web Socket Server. A WebSocket client controlled by the Display Application receives the compressed data and pushes it into a temporary concurrent queue. The Display Application then pops compressed data from the queue, decompresses, unpacks, and then draws the OpenGL calls in a render loop. A block diagram of this process is shown in Figure 23 below.

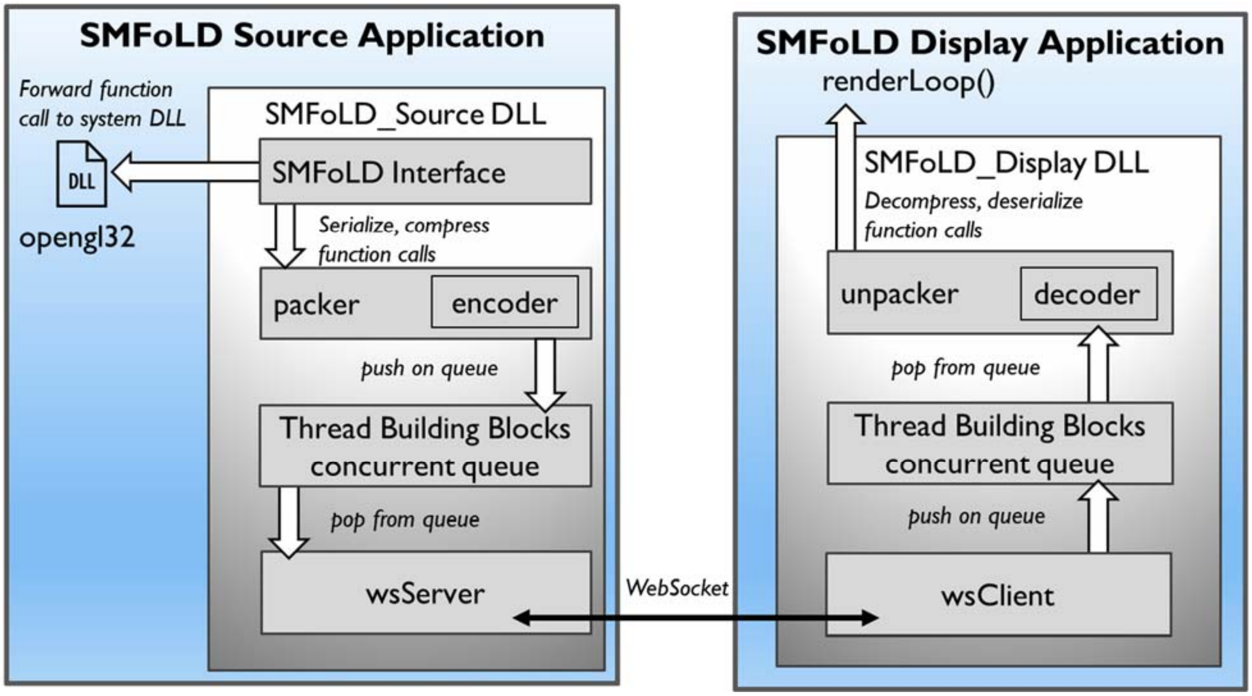

Figure 23. Simplified Block Diagram of SMFoLD State

Open source packages are used where possible within SMFoLD. SMFoLD makes use of a variety of open source libraries. Functionality, performance, code base maturity, and license compatibility were used as selection criteria (listed in no particular order). The complete library list with version number, license, and source information is displayed in Table 10 below.

Table 10. Open Source Libraries used in SMFoLD

Library	Version	License	Source
Boost C++	1.68.0	Boost v1.0	https://www.boost.org/
WebSocket++	0.8.1	BSD	https://github.com/zaphoyd/websocketpp
GLM, OpenGL Mathematics	0.9.9.3	MIT	https://glm.g-truc.net/0.9.9/index.html
MessagePack for C++	3.1.1	Boost v1.0	https://github.com/msgpack/msgpack-c
Intel Thread Building Blocks	2019 update 3	Apache v2.0	https://github.com/01org/tbb/releases
Blosc	1.15.2	BSD	https://github.com/Blosc/c-blosc

SMFoLD Project Components are as follows:

Source Application
Previously discussed (see Section 4.3.1.1)

Display Application.
Previously discussed (see Section 4.3.1.3)

Packer
The packer, based on MessagePack binary serialization, packs function parameters as an n-tuple and function pointers as a short integer opcode. The only code change since the last update was adding the encoder call to the packSwapBuffers function.

Encoder and Decoder
An encoder based on the Blosc compression framework is integrated into the existing packer. The current implementation uses Blosclz, a compressor based on FastLZ, to transmit data to the processor cache faster than the traditional, non-compressed, direct memory fetch approach via a memcpy() OS call. Blosc can accelerate memory bound computations, which is a typical need in vector-vector operations.

Thread Building Blocks
TBB provides a variety of high quality parallel and concurrent data structures that are suitable for use in production code. Currently SMFoLD uses the TBB concurrent queue. Commands are processed first in first out (FIFO) with a producer thread (packer) and a consumer thread (wsServer) concurrently accessing the queue.

Testing Code Correctness
Byte for byte data integrity was verified for MessagePack serialization and deserialization of function call parameter data. Likewise, data integrity was verified for binary strings compressed and decompressed by Blosc. The Visual Studio "Visual Leak Detector" plugin was then used to check for any memory leaks that may have been introduced during new development. Based on testing, data integrity is maintained through serialization, compression, decompression, and deserialization. Packer, encoder, decoder, and unpacker are free of memory leaks. Testing has been and will be repeated whenever tunable parameters are changed from a configuration that has been previously tested.

Testing Performance
A set of 5000 raw, uncompressed frames (approximately 90 seconds) was recorded to disk from the Test Source Application. The frames were then loaded from disk and compressed to form a second set of compressed frames. Since the Test Source Application allows for user interaction, the compression was applied to the saved files to eliminate any variation that would occur in a live run. A comparison of the raw and compressed frame sizes is shown in the Figure 24 bar graph below.

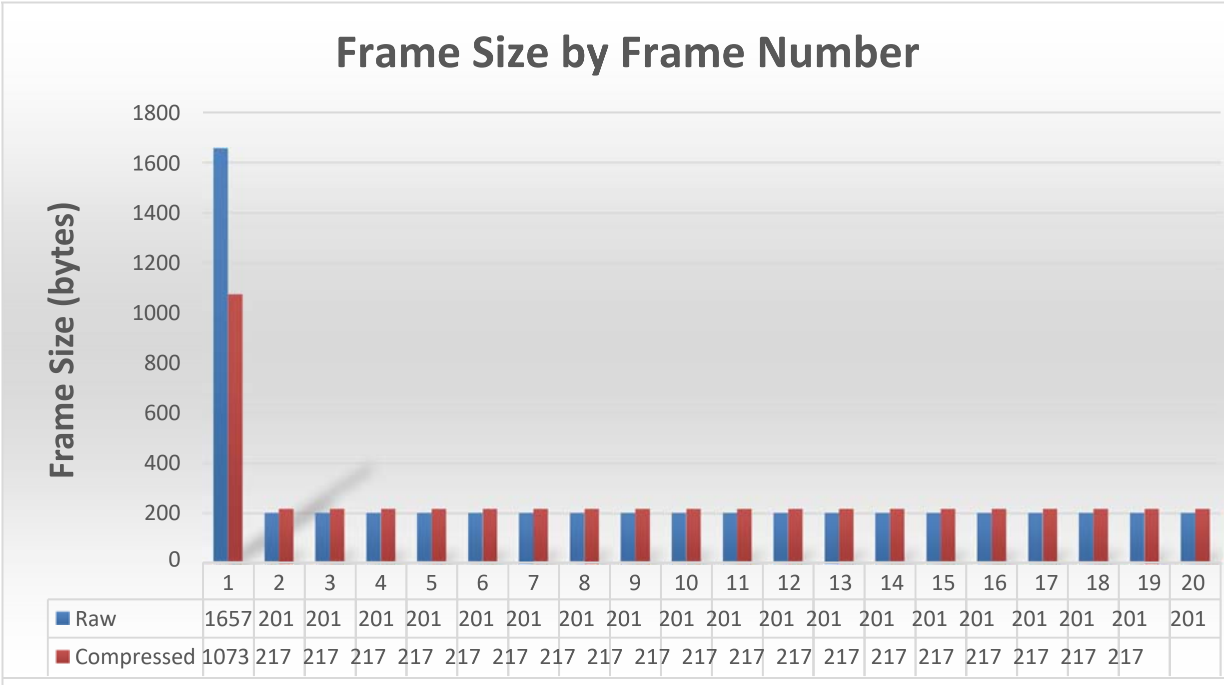

Figure 24. Frame by Frame Comparison of Raw vs. Compressed Frame Size

Based on these results, the current Test Source Application is too simple to perform any meaningful performance testing and optimization on the encoder. A new Source Application program with more complex content will need to be developed before more testing can be performed.

It has also become apparent that the current frame size profile is not representative of a real-world application. The first frame is largest because it contains all the geometry from vertex buffered objects (VBOs), vertex array objects (VAOs), and source code from all shaders. Every subsequent frame contains only matrix data for changes to the camera and/or scene states. Additional geometry, texture, and shaders will need to be loaded dynamically in the Test Source Application to provide a more realistic test case.

Though only data for 20 frames is displayed, the remaining frames in the 5000 frame test set were all the same 201 byte size. All of the frames in the test set after frame 1 are so small that compression cannot be applied efficiently. In fact, attempting compression resulted in a 16 byte increase in frame size due to compressor overhead.

4.3.6 Encryption Capability

Encryption is implemented in SMFoLD by WebSockets. The Websocket TCP/IP packet implementation encrypts the packets on the outgoing end and decrypts the ethernet packets on the incoming end.

Secure Sockets Layer (SSL v2, v3) and Transport Layer Security (TLS 1.0, 1.1) have all been deprecated due to known vulnerabilities and exploits. The National Institute of Standards and Technology (NIST) Information Technology Laboratory defines requirements and best practices for use of the TLS on government systems in Special Publication 800-52: https://csrc.nist.gov/publications/detail/sp/800-52/rev-2/archive/2017-11-15

In brief, NIST requires usage of TLS 1.2, depicted in Figure 25 below, at minimum and recommends a transition to TLS 1.3 prior to January 1, 2024: https://csrc.nist.gov/publications/detail/fips/140/2/final

The original Websocket++ implementation uses Mozilla's Modern Browser Profile to define available cryptographic algorithms. The Mozilla cipher suite list represents a balance between compatibility, computational complexity, and security. Backwards compatibility makes sense for a web browser but is unnecessary for SMFoLD. Both SMFoLD client and server endpoints are new developments and are intended to be used exclusively with each other. Restricting the cipher suites to only the subset of those that have received FIPS validation allows Websocket++ to conform to the SP 800-52 requirements. The allowable cipher suites are shown again below:

The TLSv1.2+FIPS validated cipher suites are as follows:

https://wiki.openssl.org/index.php/FIPS_mode_and_TLS
ECDHE-RSA-AES256-GCM-SHA384
ECDHE-ECDSA-AES256-GCM-SHA384
ECDHE-RSA-AES256-SHA384
ECDHE-ECDSA-AES256-SHA384

TLS 1.2 vs 1.3
Configuring Boost.Asio to reject all other connection types allows the client and server to exclusively utilize TLS 1.3 shown in Figure 26 below for encrypting the data stream. The main benefits of TLS 1.3 are better cryptographic security and a potential reduction in latency. This is achieved through the use of new keys each session and a streamlined handshake process. With the implementation of TLS 1.3, further end to end testing of the SMFoLD encryption implementation has been carried out.

A discussion of TLS 1.3 improvements over TLS 1.2 can be found at the following URL: https://www.thesslstore.com/blog/tls-1-3-handshake-tls-1-2/

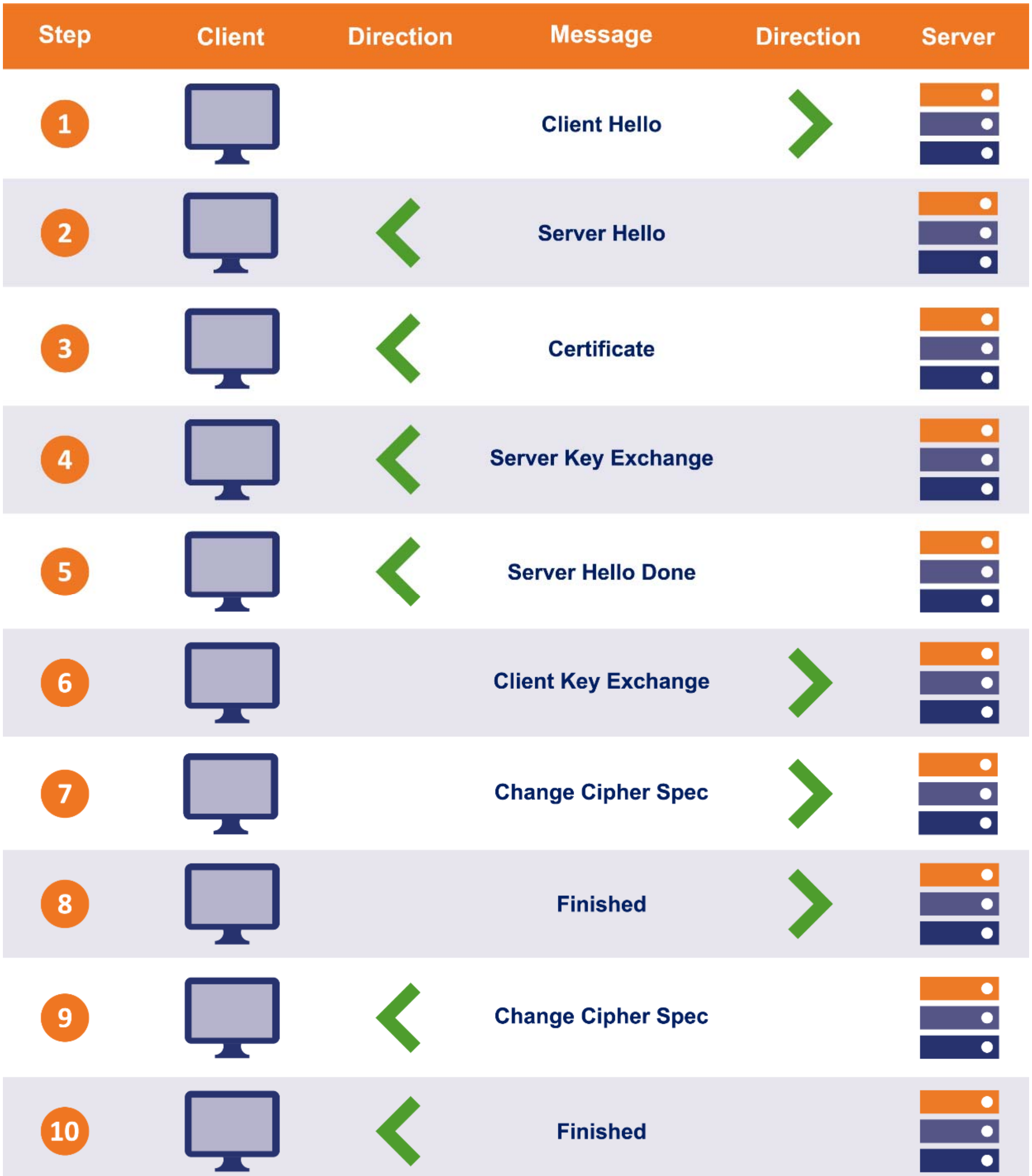

Figure 25. TLS 1.2 Handshake

Figure 26. TLS 1.3 Handshake

SMFoLD Update
Secure WebSocket client and server test modules have been updated to exclusively use TLS version 1.3 and FIPS validated cryptographic algorithms. Additional testing has been carried out and these modules have been integrated into the SMFoLD code branch.

4.3.7 Audio Capability

The current SMFoLD audio implementation allows the server to capture an audio stream and to write the captured audio stream to a local memory buffer or disk. Additional development is needed to send the audio frames over an encrypted network to a client process for decryption and rendering on the client-side endpoint device.

Code has been developed to use the Windows Audio Session Application Programming Interface (WASAPI) to target Windows 10 deployment. WASAPI allows the server to access the audio data between an endpoint buffer and an endpoint device. The server periodically reads audio data from the system's shared endpoint buffer and implements a loopback mode. This allows the server process to capture the audio stream that is being rendered by the default endpoint device of the system. This loopback functionality is necessary to support technologies such as acoustic echo cancellation, but more generally this allows capture of the entire audio engine mix of the system.

Dependencies for using Boost Beast (WebSockets for data transmission) and OpenSSL (necessary for Boost Beast) have been implemented. Code has been compiled and tested for a server and client for echoing a single message. Small changes were made to the code to change the behavior so that the server can send any number of messages to the client that are then output on the client side. The server side was adapted to capture audio data from the system and then send that instead of messages. A loopback capture example was adapted and combined to capture the audio data. The client side was modified to play the audio data received. The final steps to be taken are to add encoding and decoding of the audio data and integrate with the SMFoLD code base. Audio has not been fully implemented as of the end of this Phase II effort, but work on audio will continue as SMFoLD evolves as an open source project.

4.3.8 Display Application Feedback

4.3.8.1 Point of View Control

The OpenGL Model View Projection matrix maps points in world space to screen coordinates. The projection matrix defines the render volume and field of view. The view matrix defines the camera position and orientation. The model matrix defines any rotation, scaling, or translation that will be applied to objects in the world. Model View Projection matrix is formed by the multiplication of the projection, view, and model matrices. The general form of the projection and view matrices is shown in Figure 27 below:

$$\text{Projection} = \begin{bmatrix} \frac{1}{\text{aspect}\cdot\tan\left(\frac{fov}{2}\right)} & 0 & 0 & 0 \\ 0 & \frac{1}{tan\left(\frac{fov}{2}\right)} & 0 & 0 \\ 0 & 0 & -\frac{\text{far}+\text{near}}{\text{far}-\text{near}} & -\frac{2\cdot\text{far}\cdot near}{\text{far}-near} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

```
glm::perspective(fov, aspect, near, far)
```

$$\text{View} = \begin{bmatrix} R_x & R_y & R_z & 0 \\ U_x & U_y & U_z & 0 \\ D_x & D_y & D_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & -P_x \\ 0 & 1 & 0 & -P_y \\ 0 & 0 & 1 & -P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
glm::lookAt(CameraPos, lookAt, upVec)
```

R: right vector **U: up vector**
D: direction vector **P: camera position**

Figure 27. General Form of the OpenGL Projection and View Matrices

Point of view control is achieved by changing the values that define the projection and view matrices. Field of view can be increased or decreased to effectively zoom in or out. Likewise, changing the camera position and direction vector allows movement though the world. Anything outside the projection matrix render volume will be "clipped" and thus not visible. Note that the world actually moves around the camera rather than the camera moving though the world. This distinction makes little difference for a high-level discussion, but it is worth noting that the order in which transformations are applied is very important to achieving the expected onscreen result.

Since the matrices on the Source and Display applications can be transformed independently, there are several possible usage scenarios that emerge by specifying which matrix is passed to the shader. The hardware layout of a typical SMFoLD use case is shown in Figure 28 below.

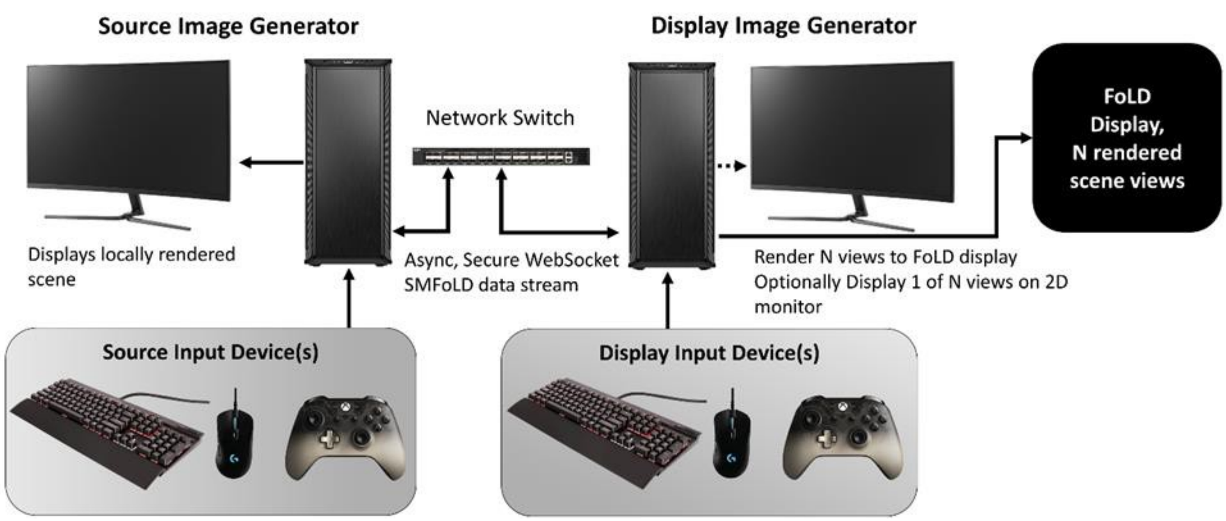

Figure 28. Typical SMFoLD Configuration using Two Workstations

The input devices connected to the Source machine can be used to manipulate the view locally. Any changes will be mirrored on the remote Display machine as the view transmitted from the Source is the basis for the N (or N x M) rendered views on the FoLD system. Input latency will be increased by the transmission time.

The input devices connected to the Display machine can be used to manipulate the view locally. In this case the local point of view will be decoupled from the Source. The Source machine will still control the render volume, but the camera on the Display Application can be moved freely inside that volume independent of the Source Application. Input latency will be low because the device(s) are connected directly to the local machine. The render volume position will need to be advanced on the Source Application whenever the Display Application nears the extents.

The Source machine could also request the current model view projection matrix from 1 of N (or N x M) views being rendered on the FoLD system. This scenario would likely be used for debugging and/or testing purposes.

Currently point of view control is implemented with only the mouse and keyboard. Support for game controllers as well as GUI managed display modes is planned.

Frame Rate Control
Frame rate control using Vsync has been completed for SMFoLD. Vsync is a GPU technology that forces the Display refresh rate to match the Source Application refresh rate. This technology must be programmed into the Source Application and enabled on the Display Application GPUs.

The frame rate of the Source Application cannot exceed the target frame rate of the Display Application unless frames are dropped by the Display Application. Modern GPU drivers commonly allow hardware synchronization to a display's native refresh rate as well as some form of frame pacing to reduce jitter. Higher end hardware such as the NVDIA 1080ti and NVIDIA Quadro RTX 5000 (used in the LightSpace demonstration at AFRL) allow synchronization to native refresh, half native, or adaptive between half native and max. Unfortunately, even half of the native refresh rate of the gaming laptop running the Source Application is still at 60Hz, three times greater than the LightSpace x1406C native refresh rate of 20Hz. If frames are not dropped the Display Application effectively replays the Source Application in slow motion. Measurements taken based on the demo revealed the following: after 1 second, the Source Application generated and sent 60 frames. At the same point in time the Display Application rendered 20 frames with 40 in the queue. The 3:1 disparity in frame rate makes the apparent time until a new user input is displayed two seconds in spite of the fact that the network transmission time is less than 1 millisecond (ms). This trend continues, growing the apparent latency linearly over time. If the Source Application sends frames faster than the FoLD can display them, frames have to be dropped or apparent latency is increased. This problem can be avoided completely with proper driver settings for FoLDs with native refresh rates that match typical monitors, but no official driver release from AMD or NVIDIA supports Vsync at less than half of the primary display's native refresh rate. Unless Vsync at arbitrary values for secondary displays is added in a GPU driver update, FoLDs with a native refresh rate <30Hz will require frames to be dropped to reduce apparent latency in their SMFoLD compliant Display Application. The sample SMFoLD Display Application have been updated to demonstrate how to do this for cases where the Source Application renders at an integer multiple of the Display Application target frame rate.

4.3.9 SMFoLD Streaming Performance

SMFoLD stream has been tested both locally in the TDT labs and over the internet using an Amazon Web Services (AWS) server over the internet. For both tests the frame rate achieved was 60 FPS. For local testing the latency between the Source Application workstation and the Display application workstation was 19 ms. For streaming a Source Application running on the AWS server to an external workstation, the latency was 100 ms. This is reasonably typical for an Internet connection and does not cause any local problems since the user is not looking at the remote Source Application.

4.3.10 Performance Tradeoffs

4.3.10.1 OpenGL Mesh and Texture

TDT chose to include only OpenGL mesh and texture and other primitives (vertexes, vectors, points, lines quadrilaterals, and general polygons). This specifically does not include video and, additionally ray-tracing for specular reflections and refraction is difficult to impossible in OpenGL. TDT did choose to implement OpenGL 3.3 core because it specifically requires buffered objects (for change frames—the objects are maintained on the Display computer and not streamed again) and is compatible with all versions of Modern OpenGL above 3.3. The OpenGL 3.3 implementation can be easily upgraded to later versions by adding the additional features implemented in the later versions, up through the present latest version which is OpenGL 4.6. For the sake of simplicity, OpenGL core 3.3 was chosen for SMFoLD—it's the simplest version of OpenGL that includes/requires buffered objects, which made it the easiest to implement in the allowable time frame. Graphics streaming (OpenGL streaming in this case) is also much more efficient than trying to stream raster data; only a single graphics frame needs to be sent and then rendered at the Display N (or N x M) times for parallax displays. These frames are typically megabytes and can also be compressed so that for instance a 1 Gbps link can transfer complex 3D frames at 60 frames per second or even much higher rates (see Table 9).

4.3.10.2 Audio

The decision was made to stream audio without synchronizing it to OpenGL frames. Microsoft WASAPI is used to capture all of the audio (at the card level) on the Source computer and stream it from the Source computer to the Display computer where the user is allowed to choose any connected component for playback (headphones, speaker, other audio card output). There are so many different audio interfaces (e.g., OpenAL, Java Sound, Web Audio, . . .) that trying to be compatible with all of them would be onerous. Capturing their calls from the Source Application would require another very large effort, similar to SMFoLD itself. OpenGL does not have an audio component, so it cannot be directly used to stream audio with the OpenGL frames. Synchronizing the audio with OpenGL frames would require time-stamping the frames and the audio output, and could lead to very choppy and unrecognizable audio. In the case where streaming frames is fast, then the audio would have some synchronization with the 3D frames. Choosing WASAPI and unsynchronized streaming allows the use of a remote narrator microphone and could eventually allow two-way communication if the choice is made at a later date to add this feature to SMFoLD. While a WASAPI application has been developed by TDT it is not presently integrated into SMFoLD.

4.3.10.3 Point to Point Streaming

Presently SMFoLD only allows streaming from a Source Computer to an SMFoLD compliant Display (which is probably a computer, although the SMFoLD compliant driver could be built into the Display hardware). In the future it is likely to become desirable to stream to multiple remote (or even on the same site) FoLD displays.

4.3.10.4 Compression

As shown in Table 8, TDT did not choose the algorithm with the highest compression, but instead chose the blosclz algorithm because of its good compression performance and excellent speed. Higher compression ratio algorithms were considerably slower and the blosclz algorithm is a good choice in terms of trading compression for speed.

4.3.10.5 Point of View Control

The input devices connected to the Source machine can be used to manipulate the view locally. Any changes will be mirrored on the remote Display machine as the view transmitted from the Source is the basis for the N (or N x M) rendered views on the FoLD system. Input latency will be increased by the transmission time.

The input devices connected to the Display machine can be used to manipulate the view locally. In this case the local point of view will be decoupled from the Source. The Source machine will still control the render volume, but the camera on the Display Application can be moved freely inside that volume independent of the Source Application. Input latency will be low because the device(s) are connected directly to the local machine. Given Display Application control of the POV, there is no necessity for POV feedback from the Display to the Source.

5.0 CONCLUSIONS

The SMFoLD model has been implemented with the creation of SMFoLD Source and Application DLL's (SMfoLD_Source.lib and SMFoLD_Display.lib). Demo Source and Display Applications have also been written, and SMFoLD has been demonstrated on two different FoLD systems, one of them being the LightSpace x1406C, 20 LCD-screen volumetric display, and the other being the LaunchTN 22-channel HP System at TDT. The LightSpace demo was held 1 Aug 2019 at AFRL and various other occasions at TDT. SMFoLD demonstration on the TDT LaunchTN FoLD system was completed on 6 December 2019.

Two workshops were held to provide information to display and application developers about the SMFoLD effort and to solicit input for further SMFoLD development. The workshops were held in association with Display Summit conferences in 2017 and 2018 (APPENDIX D). A previous workshop was held in October 2016 as part of the Phase I SMFoLD effort and was reported on in the Phase I reports.

SMFoLD was briefed at SD&A 2018 (documented in APPENDIX B), Display Summit 2017, and Display Summit 2018 conferences, AFRL FoLD workshops in 2017 and 2018 (APPENDIX E), and a SMPTE Technology Webcast in November 2019.

An SMFoLD paper was published in the IS&T International Symposium on Electronic Imaging Conference Proceedings (January, 2018) and a paper has been submitted to the SMPTE Motion Imaging Journal (probable publication in Q1 2020).

It seems probable that SMFoLD will be useful for streaming from remote creation sites for viewing by content consumers on their own FoLD systems. Several FoLD display developers have already expressed interest in the SMFoLD model.

For comparison of the SMFoLD model to other 3D streaming standards, please see APPENDIX F.

6.0 RECOMMENDATIONS

6.1 Specific Recommendations

6.1.1 SMFoLD Future Development

The current version of SMFoLD has demonstrated the viability of streaming 3D content rendered by a source computer over a local network or the internet to a remote FoLD system where multiple viewpoints of the 3D content are rendered to create a light field. This capability will allow graphics programmers, intelligence analysts, simulator instructors, and other end users to create graphical narratives that can be rendered locally and be simultaneously streamed to a remote SMFoLD compliant FoLD, stereoscopic, or 2D display. Additional development is needed to provide the capability for the end user to simultaneously stream the content from a local computer to multiple remote SMFoLD compliant displays.

The current SMFoLD model is based on OpenGL 3.3. Any OpenGL 3.3 application can be recompiled against the SMFoLD.dll to become SMFoLD compliant. Future development is needed to implement later versions of OpenGL through version 4.6 and other APIs such as Vulkan and DirectX. Such development will broaden the number of applications that can be made SMFoLD compliant and have their 3D rendered content displayed on a remote SMFoLD compliant display. Continued support is needed to develop future versions of SMFoLD based on modern OpenGL, Vulkan, DirectX and other graphics APIs.

The current SMFoLD model requires mesh and texture data types to be used for rendering. Future development is needed to allow other data types such as video, 2D plus depth, and point clouds. Continued support is needed to implement data types other than mesh and texture.

Work should be undertaken either by the SMFoLD open source support community or some other entity to implement the remainder of the OpenGL 3.3 core functions that have not been automatically generated by the TDT parsing script.

6.1.2 SMFoLD Support Organization

The SMFoLD model clearly still needs a home. Making it an open source, open standard, freely available on GitHub may lead to a volunteer developer organization supporting it. A further support structure of some form for SMFoLD needs to be found. TDT is making efforts to seek support. The Khronos Group could potentially be a good home for SMFoLD if interest can be developed. Since OpenGL is the core of SMFoLD and Khronos is the keeper of OpenGL this could potentially be a very good fit, if interest can be developed from Khronos.

The SMFoLD model is quite useful for streaming 3D OpenGL graphics scenes from one location to another. It allows for the remote location to modify POV up to the extent of the data transmitted. There is presently no model for streaming 3D graphics rendered on a local computer simultaneously to displays at remote locations. Additional work is needed to demonstrate the value of the SMFoLD 3D data streaming paradigm and to secure the long term support of a standards organization.

7.0 REFERENCES AND BIBLIOGRAPHY

1. OpenAL, http://openal.org/ , accessed 13 Jan 2018.
2. OpenAL-Soft, http://openal-soft.org/ , accessed 13 Jan 2018
3. GitHub, https://github.com/Blosc/c-blosc2, accessed on 13 Jan 2018.
4. HDFGroup Blog, https://www.hdfgroup.org/2016/02/the-blosc-meta-compressor/, accessed on 13 Jan 2018.
5. Thomas, C., "SMFoLD Streaming 3D Media", SMFoLD Workshop 2017, Rockwell Collins facility, Sterling, Virginia, http://www.smfold.org/2017-smfold-workshop/agenda/ , accessed on 15 Dec. 2019.
6. Display Summit, Oct. 4-5 2017, Rockwell Collins, Sterling Virginia, http://www.displaysummit.com/previous-events/2017_display_summit/, accessed on 15 July 2018.
7. Daniel, J., et al., "Initial Work on Development of an Open Streaming Media Standard for Field of Light Displays (SMFoLD)", IS&T International Symposium on Electronic Imaging, San Francisco, CA, V. 2081 (EI 2018).
8. Display Summit, "Updates on Development of An Open Streaming Media Standard for Field of Light Displays (SMFoLD)" Oct. 2-3 2018, Harman International, Northridge, CA, http://www.displaysummit.com/2018_display_summit/agenda/ , accessed on 15 Dec 2019.
9. Thomas, C., et al., "Status of Work on a Streaming Model for Field of Light Displays (SMFoLD) Standard", submitted to SMPTE Motion Imaging Journal, 22 November 2019.
10. 2019 Light Field and Holographic Display Summit, "Galloping to the Finish Line: Streaming Media for Field of Light Displays Standard", Oct. 8-9, 2019, CableLabs facility, Louisville, CO, http://www.displaysummit.com/2019-light-field-and-holographic-display-summit/2019-light-field-and-holographic-display-summit-agenda/ , accessed on 15 Dec 2019.
11 Thomas, C., "Streaming Model for Field of Light Displays", SMPTE Technology Webcast, 14 Nov 2019, https://www.smpte.org/technology-webcast-2 , accessed 15 Dec 2019.
12. Inces, S, et al. "Depth estimation for view synthesis in multiview video coding." *3DTV-CON*. 2007.
13. Würmlin, S, et al., "3D video fragments: Dynamic point samples for real-time free-viewpoint video." *Comput & Graph* 28.1: 3-14. 2004.
14. Humphreys, G, et al. "Distributed rendering for scalable displays." *ACM/IEEE Conf. Supercomputing*, 2000.
15. Humphreys, G, et al. "Chromium: a stream-processing framework for interactive rendering on clusters." *ACM TOG*. 21.3: 693-702. 2002.
16. Eilemann S, R Pajarola, "The Equalizer parallel rendering framework," Tech Report. Univ Zurich 2007.
17. Ilmonen, T, et al. "Broadcast GL: An alternative method for distributing OpenGL API calls to multiple rendering slaves," *J. WSCG*, 13(2), pp. 65-72, 2005.
18. Lorenz, M, G Brunnett, "Optimized visualization for tiled displays," *Eurographics*, 2006.
19. Paul, B, et al. "Chromium render server: Scalable and open remote rendering infrastructure." *IEEE TVCG* 2008.

20. Hu, S-Y. “A case for 3d streaming on peer-to-peer networks.” ACM Int’l Conf 3D Web Technology. 2006.
21. Vani, V. “New approaches for 3D streaming and rendering based on predictive modeling.” 2014.
22. Evans, A, et al. “3D graphics on the web: A survey.” *Computers & Graphics* 41: 43-61. 2014.
23. glTF. Runtime asset format for WebGL, OpenGL ES, and OpenGL. https://www.khronos.org/gltf. 2014.
24. Tanner, CC, et al. “The clipmap: a virtual mipmap.” ACM Computer graphics & Interactive Techniques. 1998.
25. Losasso, F, H Hoppe. “Geometry clipmaps: terrain rendering using nested regular grids.” *ACM TOG* 23, 2004.
26. Hoppe, H. “Progressive Meshes.” *SIGGRAPH*. pp. 99-108, 1996.
27. Kauff, P, et al., “Depth map creation and image-based rendering for advanced 3dtv services providing interoperability and scalability. *Signal Processing: Image Communication* 22, 2, 217 – 234. 2007.
28. Smolic, A, P Kauff. “Interactive 3D video representation and coding technologies” *Proc IEEE* 93:98-110. 2005.
29. Lamboray, E, et al. “Real-time streaming of point-based 3D video.” *IEEE VR*, 2004.
30. Jurgelionis, A, et al. “Platform for distributed 3D gaming.” *Intl J Comput Games Tech*. 2009.
31. Dinh, HT, et al. “A survey of mobile cloud computing: architecture, applications, and approaches.” *Wireless communications and mobile computing* 13.18: 1587-1611. 2013
32. Kruger, M. “The future of cloud gaming?” *Wired*. Aug 2013.
33. Gabor, D. “Microscopy by Reconstructed Wavefronts”, Proc Roy Soc. London Ser A, A197, p. 459, 1949.
34. Holliman, NS, et al., “Three-Dimensional Display: A Review and Applications Analysis,” *IEEE Trans Broad*, 57(2), pp. 362-371, June 2011.
35. Konrad, J, M Halle, “3-D displays and signal processing: An answer to 3-D ills?” *IEEE Sig Process Mag* 2007.
36. Page, DL, et al., “Vergence and accommodation in simulation and training with 3D displays.” *I/ITSEC*. 2014.
37. Bove, VM. “What is Holographic Television, and Will it Ever be in My Living Room?” *SMPTE Int’l Conference on Stereoscopic 3D for Media and Entertainment*, 2010.
38. Smalley, D, et al. “Holographic video display based on guided-wave acousto-optic devices”, *SPIE*, 6488. 2007.
39. Smithwick, Q, et al. “Interactive holographic stereograms with accommodation cues,” *SPIE Practical Holography XXIV*, Vol. 7619, 2010.
40. Blanche, P-A, et al., “Future of photorefractive based holographic 3D display.” *SPIE Practical Holography XXIV*, 7619, 2010.
41. Halle, M., “Autostereoscopic displays and computer graphics,” *SIGGRAPH*, 1997.
42. Slinger, C, C Cameron, M Stanley, “Computer-generated holography as a generic display technology,” *Computer*, 38(8), pp. 46-53, Aug 2005.
43. DeBitetto, D., “Holographic panoramic stereograms synthesized from white light recordings,” *Applied Optics*, 8(8), pp. 1740-1741, 1969.

44. Halle, M, "Holographic stereograms as discrete imaging systems", *SPIE*, Vol. 2176, 1994.
45. King, M, et al. "A new approach to computer-generated holography," *Applied Optics*, 9(2), pp 471-475, 1970.
46. Klug, M, et al. "Optics for Full-Parallax Holographic Stereograms," *SPIE Pract Holo XI*, 3011, pp 78-88, 1997.
47. Blanche, P-A, "Toward the ultimate 3-D display," *Information Display*, 28(3), 2012.
48. Thomas, CE, "System & Methods for Angular Slice True 3-D Display," US Pat. 7,513,623 B2, Apr. 7, 2009.
49. Thomas, CE, "System & Methods for Angular Slice True 3-D Display," US Pat. 8,075,138 B2, Dec. 13, 2011.
50. Cutting, J, "How the eye measures reality and virtual reality," Behavior Research Methods, Instruments & *Computers*, 29(1), pp. 27-36, 1997.
51. RealView Imaging Ltd., www.realviewimaging.com, Accessed on Feb. 5, 2016.
52. Balogh, T, et al. "An interactive multi-user holographic environment," SIGGRAPH, 2006.
53. Zebra Imaging, www.zebraimaging.com, Accessed on Feb. 2, 2016.
54. SeeReal Technologies, https://www.seereal.com/contact/, Accessed on Oct.14, 2017.
55. Light Field Lab, http://www.hollywoodreporter.com/behind-screen/nab-consumer-ready-holograms-move-closer-reality-997242 , accessed on 8 Sept 2017
56 LightSpace Technologies, http://lightspace3d.com, accessed on 18 Mar 2018.
57 Chinnock_Chris, https://www.insightmedia.info/lightspace-back-with-volumetric-display/ , accessed 18 Apr 2018.
58. Geng, J. "Volumetric 3D display provides true perception of objects," *SPIE Newsroom*, July 23, 2012.
59. Physical Optics Corporation, http://www.poc.com/. Accessed on Jan 29, 2016.
60. Setred, www.setred.com, Accessed on Jan 30, 2016.
61. zSpace, www.zspace.com, Accessed on Jan. 30, 2016.
62. Brennesholtz, M. "Provision looking for crowd-source funding on Kickstarter," *Display Central*, July 2013.
63 Multi-GPU Graphics Programming, AMD Fusion Developer Summit 2011 http://developer.amd.com/wordpress/media/2013/06/2115_final.pdf , accessed Aug 2018.
64 Molnar, S., Cox, M., Ellsworth, D., & Fuchs, H., "A sorting classification of parallel rendering", IEEE computer graphics and applications, 14(4), 23-32, (1994).
65 Graham, R. L., Woodall, T. S., & Squyres, J. M., "Open MPI: A flexible high performance MPI", In International Conference on Parallel Processing and Applied Mathematics (pp. 228-239), Springer, Berlin, Heidelberg, (2005, September).
66 Hernandez, B., Pérez, H., Rudomin, I., Ruiz, S., de Gyves, O., & Toledo, L, , "Simulating and visualizing real-time crowds on GPU clusters", Computación y Sistemas, 18(4), , 651-664, (2014).
67. Chinnock, C., "Introduction to SMFoLD Workshop 2017", SMFoLD Conference, October 2017, http://www.smfold.org/introduction-to-smfold-workshop-2017/ , Accessed October, 2017.
68. Hamilton, Matthew, "Light Field Displays: From Current Development to the Next Generation", SMFoLD Workshop October 2017, http://www.smfold.org/wp-

content/uploads/2017/10/Avavlon-Light-Field-Displays-From-Current-Developments-to-the-Next-Generation_RBEdits.pdf , accessed October, 2017.
69. Lude, Pete, "An Overview of Light Field Acquisition", SMFoLD Workshop, October 2017, http://www.smfold.org/wp-content/uploads/2017/10/P-Lude_Light-field-capture_ver2.pdf , accessed October 2017
70. Robertson, Sam, "Trade-offs in Light Field Streaming, Processing and Display Requirements for High and Low Fidelity Applications", SMFoLD Workshop, October, 2017, http://www.smfold.org/wp-content/uploads/2017/10/Holochip-SMFoLD-Workshop-2017-.pdf , accessed October 2017.
71. Manniquis, Nilo, "Improving Battlespace Awareness, Reducing Warfighter Workload, and Enabling Rapid Response Through the Use of Collaborative 3D Holographic Display", SMFoLD Workshop, October 2017, http://www.smfold.org/wp-content/uploads/2017/10/SMFOLD-2017-Nilos-Brief.pdf , accessed October 2017.
72. Daniel, Jamison, "Visualization Technologies and the Challenge of High Performance Computing at the Oak Ridge Leadership Computing Facility", SMFoLD Workshop, October 2017, http://www.smfold.org/wp-content/uploads/2017/10/ORNL_SMFoLD_workshop_talk_2017-V3.pdf , accessed October 2017.
73. Husak, W., "JPEG-PLENO's Interest in Light Field Images" , SMFoLD Workshop, October 2017, http://www.smfold.org/wp-content/uploads/2017/10/JPEG-Pleno-for-SMFoLD.pdf , accessed October 2017
74. Thomas, C., "SMFoLD Streaming 3D Media", SMFoLD Workshop, October, 2017, http://www.smfold.org/wp-content/uploads/2017/09/TDT-SMFoLD-Workshop-20171003-Distribution-A.pdf , accessed October, 2017.
75. Burnett, T., "Object Graphics Library (ObjGL) and the Heterogeneous Display Environment", http://www.smfold.org/wp-content/uploads/2017/09/FoVI3D_ObjGL_Intro.pdf , accessed October 2017.
76. Hinds, Arianne, "Toward the Realization of 6 DoF Standards, http://www.smfold.org/wp-content/uploads/2017/10/ATHinds-final.pdf , accessed October 2017.
77. Karafin, J., "Holographic Content Considerations", http://www.smfold.org/wp-content/uploads/2017/10/LightFieldLab_DisplaySummitSummaryPresentation_10032017-1.pdf , accessed October 2017.
78. Chinnock, Chris, "SMFoLD Panel Discussion", SMFoLD Workshop Summary Report, Private Document, October 2017.
79. "5 Common Encryption Algorithms and the Unbreakables of the Future", http://www.storagecraft.com/blog/5-common-encryption-algorithms, Accessed January 2017
80. Annen, T, et al. "Distributed rendering for multiview parallax barrier displays," *SPIE*, Vol. 6055, 2006.
81. Barnes, MJ. "The human dimension of battlespace visualization: Research and design issues." No. ARL-TR-2885. ARL Aberdeen Proving Grounds, 2003.
82. Chun, W-S, et al., "Spatial 3-D infrastructure: Display-independent software framework, high-speed rendering electronics, and several new displays," *SPIE*, Vol. 5664, p. 302, 2005.
83. Bove, VM. "Engineering for live holographic TV." *Motion Imaging Journal, SMPTE* 120.8: 56-60. 2011.

84. Dixon, S, et al. “Human Factors Guidelines for Applications of 3D Perspectives: A Literature Review,” *SPIE*, Vol. 7327, 2009.
85. Doerr, K-U, F Kuester. “CGLX: a scalable, high-performance visualization framework for networked display environments." *IEEE Trans Vis Comput Graphics.* 17.3: 320-332. 2011.
86. Haussler, R, et al., “Large realtime holographic displays: from prototypes to a consumer product,” *Proc. SPIE Stereoscopic Displays and Applications XX*, Vol. 7237, 2009.
87. Hoffman, D. et al. “Vergence–accommodation conflicts hinder visual performance and cause visual fatigue,” *J Vision*, 8(3):33, pp. 1–30, 2008.
88. Jolly, S, et al. “Computational architecture for full-color holographic displays based on anisotropic leaky-mode modulators.” *SPIE OPTO*, 2014.
89. Klug, M, et al. “A Scalable, Collaborative, Interactive Light-field Display System.” *SID Symp. Digest of Technical Papers*, pp. 412-4. 2013.
90. Lucente, M, TA Galyean. “Rendering Interactive Holographic Images,*” ACM SIGGRAPH*. pp. 387-394, 1995.
91. McIntire, JP, et al. “What is 3D good for? A review of human performance on stereoscopic 3D displays.” *SPIE DS&S*, 2012.
92. Onural, L, et al. “Digital holographic 3D video displays.” *Proc. IEEE*: 576-589. 2011.
93. Patney, A, et al. “Piko: A framework for authoring programmable graphics pipelines.” *ACM TOG*, 34(4). 2015.
94. Patterson, R, “Human factors of 3D displays,” *J. SID*, 15, pp. 861-871, 2007.
95. Penner, E, et al. “A GPU cluster without the clutter: A drop-in scalable programmable-pipeline with several GPUs and only one PC,” *Symp. I3D*, 2006.
96. Reichelt, S, et al. “Depth cues in human visual perception & their realization in 3D displays.” *SPIE DS&S* 2010.
97. Scechrist, S. "TVs, 3-D, and holograms at display week 2014." *Inf. Disp. Mag. SID* 30: 14-18. 2014.
98. Staadt, OG, et al. “A survey and performance analysis of software platforms for interactive cluster-based multi-screen rendering.” *ACM Workshop Virtual Environ*. 2003.
99. Thomas, LC, et al. “Immersion and battlefield visualization: Frame of reference effects on navigations tasks and cognitive tunneling.” Tech Report: ARL-99-3/FED-LAB-99-2, 1999.
100. Tolcott, MA. “Understanding and aiding military decisions.” *27th Int’l Applied Military Psy Symp.* 1992.
101. Ware, C, G Franck. “Evaluating stereo and motion cues for visualizing information nets in three dimensions.” *ACM Trans Graphics*. 15.2: 121-140, 1996.
102. Watkins, WR, et al. “Search and target acquisition: single line of sight versus wide baseline stereo.*” Optical Engineering.* 40.9, pp. 1914-1927, 2001.
103. Wickens, CD, et al. “Frames of reference for displays of battlefield information: Judgment-display dependencies.” Human Factors, 42, 660-675, 2000.
104. WebSockets Protocol: https://tools.ietf.org/html/rfc6455 , accessed Jan 2018.
105 https://msdn.microsoft.com/en-us/library/bb288454.aspx, accessed 15 Aug 2018.
106. http://blosc.org/pages/blosc-in-depth/, accessed April 2018.
107. https://code.facebook.com/posts/1658392934479273/smaller-and-faster-data-compression-with-zstandard/, accessed April 2018.
108. https://github.com/Blosc/c-blosc, accessed April 2018.

109. https://github.com/facebook/zstd, accessed April 2018.
110. https://tools.ietf.org/html/rfc5246, accessed April 2018.
111. http://www.cryptlib.com/, accessed April 2018.
112. https://github.com/awslabs/s2n, accessed April 2018.
113. Alexander T. Adams and Celine Latulipe, 2013. Survey of audio programming tools. In CHI '13 Extended Abstracts on Human Factors in Computing Systems (CHI EA '13). ACM, New York, NY, USA, 781-786.
114. Guy Somberg. 2017 "Game Audio Programming in C++". Cpp Conference 2017. Bellevue, Washington USA.
115. David Gouveia. 2013. Getting Started with C++ Audio Programming for Game Development (1st ed.). Packt Publishing.Gouveia, David 26 August 2013
116. About WASAPI, https://goo.gl/X9cQPH, accessed Feb 2018.
117. Core Audio Overview, https://goo.gl/t3DLeH, accessed Feb 2018.
118. The Open Sound System, http://www.opensound.com, accessed Feb 2018.
119. The Alsa project, https://www.alsa-project.org/main/index.php/Main_Page, accessed Feb 2018.
120. FMOD, https://www.fmod.com/ , accessed Feb 2018
121. Audio Kinetic Wwise, https://www.audiokinetic.com/products/wwise/, accessed Feb 2018.
122. CRI Middleware ADX2 https://www.criware.com/en/products/adx2.html , accessed Feb 2018.
123 Lafruit, G. et al. 2018 "MPEG-I Coding Performance in Immersive VR/AR Applications" https://www.ibc.org/create-and-produce/mpeg-i-coding-performance-in-immersive-vr/ar-applications/3338.article
124 Preda, M., 2017. Report on PCC CfP answers. ISO/IEC JTC1/SC29/WG11. MPEG2017/N17251. October, 2017. Macao, China.
125 MPEG 3DG, 2017. Call for Proposals for Point Cloud Compression V2. ISO/IEC JTC1/SC29/WG11. MPEG2017/N16763.
126 Schelkens, P., et al., "JPEG Pleno: Providing Representation Interoperability for Holographic Applications and Devices", ETRI Journal, Volume 41, Issue 1, pp 93-108, 2019.
127 Ebrahimi, T.E., Foessel, S.,. Pereira, F., and Schelkens P., "JPEG Pleno: toward an efficient representation of visual reality", IEEE Multimedia, Volume 23, Issue 4, pp 14-20, 2016.
128 Schelkens, P., et al., "JPEG Pleno light field coding technologies", Applications of Digital Image Processing XLII, SPIE pp 391- 401, 2019.
129 "Information technology — JPEG 2000 image coding system – Part 1: Core coding system." International Standard ISO/IEC IS-15444-1 (2016).
130 Perra, C., et al., "Performance analysis of JPEG Pleno light field coding", Applications of Digital Image Processing XLII, SPIE pp 402 – 413, 2019.
131 ITMF Scene Graph Specification, http://immersivealliance.org/download/download-itmf-scene-graph-specification-0-9/, accessed on 7 Dec 2019.
132 "First-ever Demonstrations of the ITMF Specification conducted during Display Summit". http://immersivealliance.org/2019/10/29/light-field-and-holographic-display-summit-hosts-itmf-demonstrations/, accessed on 7 Dec 2019.

APPENDIX A – INTELLECTUAL PROPERTY RESULTING FROM THIS PROGRAM

A-1 Know How

The TDT team developed a significant body of knowledge related to high performance computing, parallelized rendering, secure networking, data transmission, graphics programming, and 3D streaming standards activities during the performance of this program. Much of this knowledge will be shared with the 3D display and applications development community in the implementation and furtherance of SMFoLD.

A-2 Trade Secrets

Any trade secrets consisting of formulas, patterns, compilations, programs, devices, methods, techniques, or processes related to SMFoLD and produced during the performance of this program will be made available to the public to the extent that such trade secrets facilitate the creation, use, and propagation of the SMFoLD standard.

A-3 Trademarks

The SMFoLD logo shown in Figure is a trademark of TDT that will be transferred to the standards organization that will be responsible for maintaining the SMFoLD standard.

Figure A.1. SMFoLD Logo

A-4 Copyrights

The SMFoLD software code and SMFoLD website produced during this program are protected by copyright. It is the intent of the TDT team and the Air Force that the code and associated documentation will be available to the public as Open Source once the code has been approved for release by the Air Force. The code will be published on GitHub and available under an Open Source license. The website and associated copyrighted content will be transferred to the standards body that will maintain the SMFoLD standard.

A-5 Patents

No patentable inventions were produced during the performance of this program

APPENDIX B - SMFOLD PAPERS AND PRESENTATIONS

B-1 SD&A Conference Overview and Paper Presentation

A paper has already been briefed at the SD&A 2018 conference on 29 January 2018.[7] The paper will be published in the Conference Proceedings and posted online by the conference organizers (http://www.stereoscopic.org/proc/index.html). The paper and associated presentation are also posted on the SMFoLD website (http://www.smfold.org/smfold-presentation-at-electronic-imaging-2018).

B-1.1 Overview

The SD&A conference is a subconference of the annual Electronic Imaging conference. The focus is on 3D displays with an emphasis on stereo based 3D. The conference provides an opportunity for researchers to present their latest developments in the field and learn about what may be coming in the future. The conference covers hardware, software, and content.
Steve Kelley from TDT attended the SD&A conference held at the Hyatt Regency San Francisco Airport Hotel in Burlingame, CA on January 29 thru 31, 2018. Mr. Kelley presented a paper titled "Initial Work on Development of an Open Streaming Media Standard for Field of Light Displays (SMFoLD)" to an audience of approximately 30 people.

B-1.2 Session 1: Stereoscopic Developments

Benjamin Backus, et al. "Use of VR to Assess and Treat Weaknesses in Human Stereoscopic Vision" Backus explained the process and efficacy of using VR and stereo to correct stereoscopic vision problems in humans. The technique uses different images in the right and left eyes in a game environment so that both eyes have to work together to master the game. Amblyopia and stereo processing deficiencies in the brain may be treated using this technique. They report encouraging results and are continuing to investigate this approach.

Ryo Kodama, et al. "Emotional Effects of Car-Based Motion Representations with Stereoscopic Images" Kodama described experiments that attempt to measure an emotional response to combining stereo 3D with motion cues. The results indicate that for certain motions the combination of motion cues and 3D visuals dramatically enhances the users' experience while some motion/stereo pairing did not. The greatest effects from combining the cues were in stopping and jumping motions.

Ayaka Sano, et al. "Mid-Air Imaging Technique for Architecture in Public Space" Sano presented a technique to create signage that appears in space separated from the background plane on buildings. The method uses guided light to create multiple images so that the light reflected from the surface converges so that as people approach the image it appears to float in space. The applications are mainly signage and kiosk applications. The system does not work well in bright light so applications are limited.

Hideo Saito, et al. "A Refocus-Interface for Diminished Reality Work Area Visualization" Saito made a presentation on using VR to allow workers to see objects occluded by tools so that workers' hands and tools appear to become translucent. The system uses multiple cameras to capture a work surface and project it through VR glasses so that the hands and tools appear to become translucent. The effect is quite stunning. This innovation is designed to improve productivity and safety in manufacturing settings.

B-1.3 Session 2: Autostereoscopic Displays 1: Light-field

Jamison Daniel, et al. "Initial Work on Development of an Open Streaming Media Standard for Field of Light Displays (SMFoLD)" Steve Kelley from TDT presented the paper. Kelley focused on the need for a streaming standard for FoLD systems and the work done so far to that end. There was also discussion about TDT's proposed approach. There were two audience questions and both focused on the problem and proposed solution to the use of shaders. The questioners did not understand how the shaders would be handled. Kelley explained that the applications developers will have to include named variables and that the display application is responsible for managing the multiple views as needed by the display. In essence the proposed approach will provide a stream of OpenGL 3D frames transmitted over a network.

Weitao Song, et al. "Simulation Tools for Light-Field Displays Based an a Micro-Lens Array" Song demonstrated software tools to simulate light fields for measuring the depth of field for FoLD displays using micro lens arrays. The project is aimed at providing tools to help FoLD developers predict image quality and aid in determining depth of field during the design process.

Hiroaki Yano, et al. "Full-Parallax Spherical Light Field Display Using Mirror Array" Yano showed a new approach for producing full parallax objects. The display uses a sphere of tiled mirrors. The sphere spins on a horizontal axis with a high speed projector illuminating it. The tiles are angled and spaced such that the light entering each eye is from a different direction. The resolution is very low and the system is limited by the update rate of the projector. The use of moving parts reduces the future outlook for usefulness of this technology.

Yu Zhao, et al. "Fast Calculation Method for Full-Color Computer-Generated Hologram with Real Objects Captured by a Depth Camera" Zhao presented their work on developing a fast method of creating computer generated holograms from objects captured using a depth camera. The objects were converted to meshes and textures but the algorithm had problems with areas that were occluded from the depth signal. The occlusion problem is no different than other methods for rendering depth camera images, however their calculation methods are faster than current published methods. They are working to perfect their methods.

Suren Vagharshakyan, et al. "Conversion of Sparsely-Captured Light Field into Alias-Free Full-Parallax Multiview Content" Vagharshakyan presented work being done to create alias free multi view images from sparsely captured data. Slides in the presentation showed before and after images of the conversions and showed that the conversion works successfully.

B-1.4 Opening Plenary

Dr. Greg Corrado "Overview of Modern Machine Learning and Deep Neural Networks - Impact on Imaging and the Field of Computer Vision" Corrado gave an overview of machine learning and deep neural networks and their application for computer vision. Corrado is a co-founder of Google Brain. Deep machine learning is moving forward at a rapid pace. Corrado believes the reason for the rapid advancement is not that people are just now learning how to implement machine learning; rather it is because the required computing power has only recently become available. Slides shown during the presentation demonstrated a high degree of accuracy by computers in identifying and labeling objects even in scenes with overlapping items. A test with computers vs oncologists showed that computers were better at detecting malignancies than humans. One slide showed a baby cuddling a teddy bear while lying on a couch. A computer was able to create a caption for the image in a manner closely resembling a human created caption.

B-1.5 Session 3: Stereoscopic Applications: VR to Immersive Analytics in Bioinformatics 1 (Joint Session)

Hua Wong, et al. "Mesoscopic Rigid Body Modeling of the Extracellular Matrix's Self-Assembly" Wong discussed the difficulty of modeling cellular activity at the mesoscopic level. Earlier attempts at modeling the inner workings of a cell required use of supercomputers. The computing requirements stemmed from modeling the physics of sub-components in a molecular chain. A method of simplifying the component objects to allow researchers to view animated models was presented. Using rigid bodies for molecular chain components reduces the computational requirements enough that cell activity can be modeled using modern desktop computers. An animation of the activity of the basement cell membrane was presented showing the various molecules interacting with each other.

Mikael Trellet, et al "Semantics for an Integrative and Immersive Pipeline Combining Visualization and Analysis of Molecular Data" Trellet showed the analysis of cell structures using a CAVE display environment. The 3D visualizations provided by the CAVE display were critical to understanding cell structure and inner workings. The presentation included 3D visuals of a cell to illustrate the effectiveness of viewing the data in 3D.

B-1.6 Session 4: Autostereoscopic Displays 2: Volumetric, Integral, Stackable, and Holographic

Shreya Patel, et al. "Recent Progress in Volumetric 3D Digital Light Photoactivatable Dye Displays" Patel presented a novel approach for creating a volumetric display using bioluminescence. A bioluminescent material is dissolved in a solution and activated with light energy. Two scanning beams are directed into the solution and light is emitted at the point where the beams intersect. A simple single color full parallax image was shown. There does not seem to be a good use case for the technology at this time. However, there may be discoveries that make bioluminescence useful in the production of displays. They are continuing to investigate the usefulness of this approach.

Hironobu Gotoda "Constructing Stackable Multiscopic Display Panels Using Microlenses and Optical Waveguides" Gotoda discussed his method for using optical wave guides and microlenses with stackable multiple panels to create a full parallax, relatively thin display. The focus of the work was to reduce light scatter so that the light path between panels forces the light to exit the lenses in the correct direction. Gotoda is looking for someone to manufacture displays but the manufacturing process will be difficult because of the placement of the wave guides which consist of fiber optic strands.

Hong Hua "Angular and Spatial Sampling Requirements in 3D Light Field Displays" Hua's presentation focused on data requirements when sampling for light field displays. The number of cameras, camera angles, camera spacing, and other metrics are being studied to find the best mix for capturing 3D scenes.

Avideh Zahkor "EI Plenary Session 2: Fast automated 3D Modeling of Buildings and Other GPS Denied Environments" Zahkor talked about the need for mapping the insides of buildings. She and her team have developed hardware and software that allows mapping of GPS denied areas using a light weight device in a backpack or mounted on a robot. The system scans for geometry and takes photos to create mesh and texture maps of the interior of buildings. These images could be useful for emergency responders, real estate agents, and others who need to know the layout of a building's interior.

B-1.7 Discussion

There continues to be work to find an inexpensive and usable technology to create light field and volumetric displays. The methods presented here for display technology either appear to be dead ends or a long way from commercialization. There is also substantial research in improving 3D data capture and 3D data generation methods. Software tools are being developed to help display researchers create and test new designs.

The presentations that did not deal with hardware mostly dealt with the importance of 3D visualization in medical research. The importance of immersive displays for 3D medical imaging was emphasized. The evidence suggests that much more can be learned from the visualization of cell models using immersive 3D displays.

B-2 SMFoLD Workshop I - 3 October 2017

The SMFoLD project was briefed at the SMFoLD Workshop in Chantilly, VA, on 3 October 2017 (discussed in D-1).[5]

B-3 Display Summit Conference - 4 October 2017

The SMFoLD project was briefed at the Display Summit Conference in Chantilly, VA on 4 October 2017.[6]

B-4 AFRL FoLD Workshop - 15 November 2017

The SMFoLD project was briefed at the AFRL FoLD Workshop at WPAFB in Dayton, OH on 15 November 2017 (discussed in E-1).

B-5 SMFoLD Workshop II - 2 October 2018

The SMFoLD project was briefed at the Display Summit/SMFoLD Workshop at the Harman International facility in Northridge, CA on 2 October 2018 (discussed in D-2).[8]

B-6 AFRL FoLD Workshop – 14 November 2018

The SMFoLD project was briefed at the AFRL FoLD Workshop at WPAFB in Dayton, OH on 14 November 2018 (discussed in E-2).

B-7 2019 Light Field and Holographic Display Summit – 9 October 2019

An SMFoLD briefing has been presented at the 2019 Light Field and Holographic Display Summit (CableLabs, Louisville, CO, 9 Oct 2019).[10]

B-8 AFRL FoLD Workshop – 14 November 2019

The SMFoLD project was briefed at the AFRL FoLD Workshop at WPAFB in Dayton, OH on 13 November 2019 (discussed in E-3).

B-9 SMPTE Technology Webcast – 14 November 2019

Charts for a SMPTE Technology Webcast have been approved by AFRL and the Webcast was presented on 14 Nov 2019.[11]

B-10 SMPTE Motion Imaging Journal

An SMFoLD paper for has been approved by AFRL and submitted for publication to the SMPTE Motion Imaging Journal.[9]

B-11 AFRL Briefings

The SMFoLD project was briefed at twelve AFRL Phase II review meetings between 25 January 2017 and October 31 2019.

APPENDIX C – PROFESSIONAL PERSONNEL ASSOCIATED WITH THIS PROGRAM

Third Dimension Technologies

C.E. (Tommy) Thomas, PhD, Chief Technical Officer	Principle Investigator
Del (Odie) Barstow, Senior Software Engineer	SMFoLD Software Development
Chris Honsinger, Senior Software Engineer	Calibration Software, Audio
Steve Kelley, Senior Software Engineer	TitaniumGL Development
David Page, PhD, Software Architect	SMFoLD Concept Development
Andrew Smith, Physics Support	Setup and Maintain FoLD Systems
Paul Jones, Business Development	Reports, Presentations, Publications

Oak Ridge National Laboratory
Advanced Data and Workflows Group
Oak Ridge Leadership Computing Facility (OLCF)

Daniel Jamison, Computer Scientist	SMFoLD Encryption
Benjamin Hernandez Arreguin, Computer Scientist	Parallelized GPUs, Compression
Mallikarjun Shankar, Group Leader	Administration & Communication
Shivam Patel, Intern	SMFoLD Audio

Insight Media

Chris Chinnock, Founder and President	SMFoLD Workshop Organization

Air Force Research Laboratory
Sensory Interface Development Section
Airman Systems Directorate
711 Human Performance Wing

Darrel Hopper, PhD, Work Unit Manager	Technical Point of Contact
Fred Meyer, PhD	Technical Expert
Eric Heft	Technical Expert

MITRE

Alexander Enzmann, Principal Engineer	Air Operations Center Expert

APPENDIX D - SMFoLD WORKSHOPS

D-1 TDT SMFoLD Workshop 1 – 3 Oct 2017, Rockwell Collins, Sterling, VA

TDT hosted an SMFoLD conference on 3 October 2017, on the day preceding the Display Summit Conference on the FoLD and 3D ecosystem, 4 and 5 Oct 2017. About 46 industry leaders signed up for the workshop and attendance was high.

Figure D.1 below shows the logos of the various partners in the SMFoLD Workshop.

Figure D.1. Partners in the SMFoLD Workshop

The workshop was designed to profile the status of light field acquisition, display, streaming and interface technology as well as standards activities in these areas. The complete presentations are available on the SMFoLD website. See below for a report on the TDT SMFoLD 3 October 2017 Workshop.

D-1.1 Introduction

Chris Chinnock of Insight Media gave an introduction and served as moderator. The introduction presented an overview of the purpose of the workshop and the issues identified by the primary sponsor, the Air Force Research Laboratory (AFRL). The attendees were asked to consider how to address these issues, such as the lack of a light field streaming media standard and how best to facilitate development.[67]

Insight Media's Chris Chinnock kicked off the meeting with an overview presentation that began with some logistics and links to access the agenda and presentations that are freely available for download on the website. The attendees were asked to consider how to address these issues, such as the lack of a light field streaming media standard and how best to facilitate development.[67]

http://www.smfold.org/2017-smfold-workshop/smfold-2017-presentations/

He continued by noting that the workshop was made possible because of a Phase II STTR award from the Air Force Research Lab (AFRL) awarded to Third Dimension Technologies with help from Oak Ridge National laboratories and Insight Media.

The contract is to develop a display agnostic proposal for a streaming media standard for light field display data. Such a standard is necessary because of the increasing amounts of 3D data in government and military applications, the need for better visualization and decision making and the lack of a standard method to format and deliver data to a range of 2D, 3D and light field displays.

The objectives of the workshop were to provide an overview of the current light field display acquisition and display ecosystem, discuss military and government applications for FoLD displays, review relevant activities of standards bodies introducing three proposals for a streaming media standard, and discuss these proposals in an open panel discussion.

Feedback from participants indicates these objectives were met. The workshop provided a unique gathering of experts in this field and progress was made in understanding the needs of end users and how the various streaming proposals will operate and potentially meet the needs of these military/government end users. Ultimately the goal of the effort is to transfer one or more proposed standards to conventional standards bodies for further deliberation, validation and standardization.

The workshop agenda is shown in Table D1.

Table D1. Workshop Agenda

Company	Contact	Abstract	start
Insight Media	Chris Chinnock	Introduction to SMFoLD Workshop 2017	9:00 AM
Avalon Holographics	Matthew Hamilton	Light Field Displays: From Current Developments to the Next Generation	9:15 AM
Mission Rock Digital	Pete Lude	An Overview of Light Field Acquisition	10:00 AM
coffee break			10:30 AM
VIZrt movie		Video of advanced 3D modeling and augment reality visualization in a broadcast environment	10:50 AM
Holochip Corporation	Sam Robertson	Trade-offs in Light Field Streaming, Processing and Display Requirements for High and Low Fidelity Applications	10:55 AM
Naval Sea Systems Command	Nilo Maniquis	Improving Battlespace Awareness, Reducing Warfighter Workload, and Enabling Rapid Response Through the Use of Collaborative 3D Holographic Display	11:20 AM
Oak Ridge Leadership Computing Facility (OLCF)	Jamison Daniel	Visualization Technologies and the Challenge of High Performance Computing at the Oak Ridge Leadership Computing Facility	11:45 AM
Lunch			12:10 PM
JPEG-PLENO	Walt Husak	JPEG-PLENO's Interest in Light Field Images	1:10 PM
TDT	Tommy Thomas	SMFoLD Streaming 3D Media	1:35 PM
FoVI3D	Thomas Burnett	FoVI3D's Display Agnostic Application Interface/Scene Description Proposal	2:00 PM
MPEG-i/ OTOY	Arianne Hinds	MPEG's Efforts to Standardize the ORBX format	2:25 PM
Light Field Labs	Jon Karafin	Benefits of ORBX for Light Field Workflow and Display	2:50 PM
coffee break			3:15 PM
Panel Discussion			3:35 PM
Closing			4:35 PM

D-1.2 Matthew Hamilton, Avalon Holographics, "Light Field Displays: From Current Developments to the Next Generation"

Matthew Hamilton from Avalon Holographics was asked to provide an overview of the various light field display approaches and their status of development.[68] He started with some definitions of light field displays and described one of the limitations of 3D light field displays – the resolution depends upon the distance from the image plane (i.e. the depth of field is often small with resolution decreasing further away from the image plane leading to blurry images, see Figure D.2. Illustration of How the Angular Resolution Affects the 3D Resolution, see Figure D.2). Increasing depth of field requires increasing the angular resolution or light rays per degree.

Hamilton then suggested the concept of a "Turing Test" for the 3D image: "A person views input that comes either from a direct view of the real world or from a simulated view of that world presented on a display. He or she has to decide: real or display?" Unfortunately, no 3D displays can pass this Turing test yet for a number of reasons. This includes limited depth of field, lack of focus cues, small field of view, low spatial resolution and various artifacts Figure D.2 below illustrates the effects of angular resolution on 3D resolution.

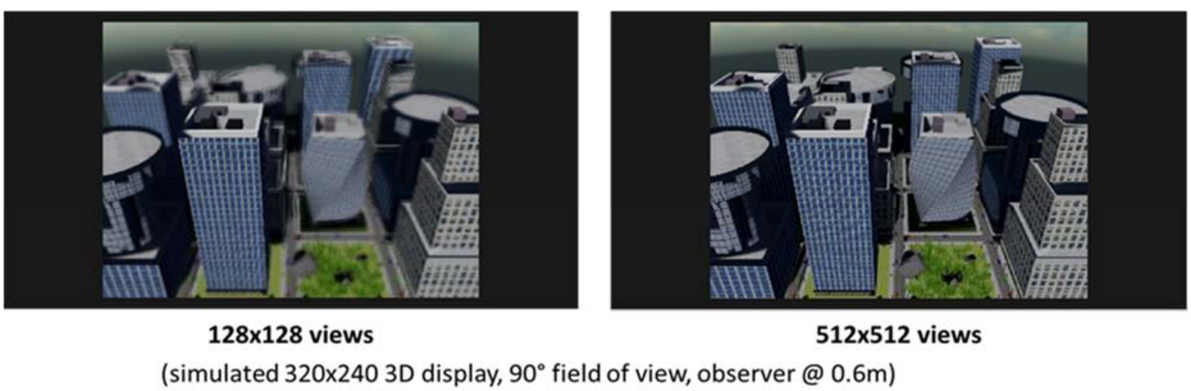

Figure D.2. Illustration of How the Angular Resolution Affects the 3D Resolution

Hamilton reviewed various classes of advanced 3D displays. He started with discussing head-mounted displays and asking if light field displays are really useful here. Systems that offer adaptive focus or multi-planar images may be good enough, he thinks, although this may not provide correct focus cues. Later in the day, Jon Karafin from Light Field Labs suggested that because of the size of the eye's pupil, it will be nearly impossible to create a light field image using a near to eye display solution.

He then described the refractive Integral Imaging approach which has an inherent trade-off between spatial and angular resolution (it is hard to have both). The approach also suffers from limited field of view and depth of field. Vast numbers of extremely tiny pixels are needed to achieve images with good FOV, depth of field (DoF), and spatial resolution.

These trade-offs and requirements make stand-alone displays challenging, but the limited FOV in HDMs may make the approach more interesting here, however. Nevertheless, high pixel density in larger-sized displays will be needed for decent image quality.

Several versions of multi-layer approaches have been developed as well. The so-called tensor approach is based upon the superposition of light rather than the direct representation of each light ray. By using temporal modulation and attenuation in these two layers, the design can overcome the spatial-angular trade-off. However, the FOV is still limited by the directional backlight layer and it is not suitable for interactive content and maybe not video content either.

Volumetric displays can also be created using a temporal multiplexing approach using a fast DLP projector and a spinning screen of some sort. These are horizontal parallax only with limited FOV. Hamilton seemed worried about the calibration and longevity of such systems.

In what Hamilton describes as the diffractive approach, the idea is to create wavelength-scale diffractive grating in a backlight layer to direct light for each pixel into several directions. This approach can allow for wide FOV options or a small eyebox with the ability to switch between 2D and 3D modes. The limitation is that the gratings cannot scale to very tiny pixel sizes so resolution will be limited (but maybe acceptable depending upon the application).

Volumetric displays use multiple layers in an additive fashion (see Figure D.3) to create an “image volume.” The big drawback of the approach is that only transparent images can be produced so there is no occlusion, no specular highlights or other effects. Medical visualization or situational awareness may be a valid use for such displays, although they would fail the Turing test.

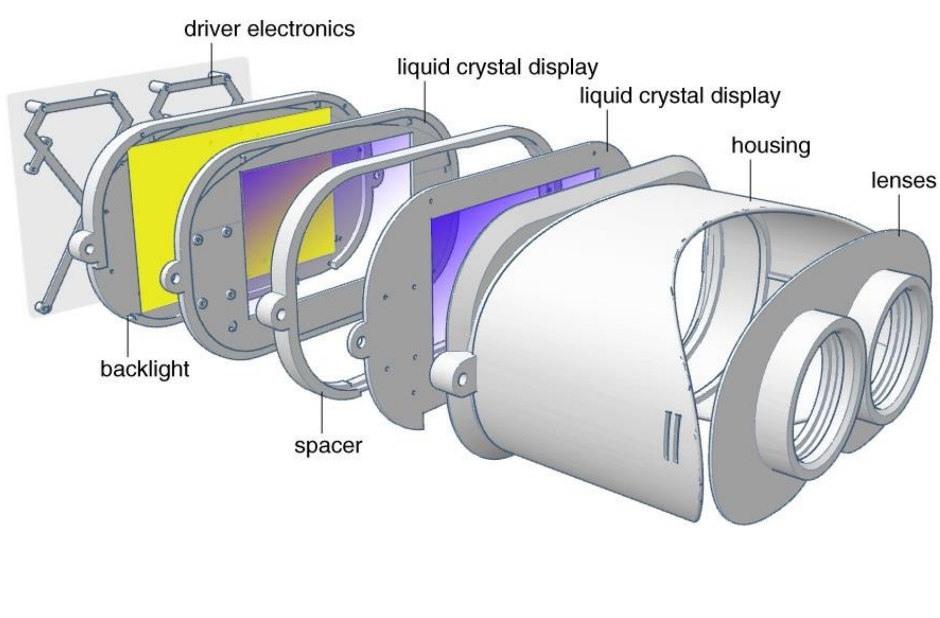

Figure D.3. Illustration of a Head Mounted Volumetric Display

For other projection solutions, such as proposed by Holografika and Third Dimension Technologies, large FOV or large eyebox designs are possible. More than 80 views in horizontal parallax only systems have been demonstrated. Scaling to full parallax is possible but increases the size and data requirements significantly.

See Figure D.4 for scaling of resolutions with distance of virtual plan from middle of 3D display.

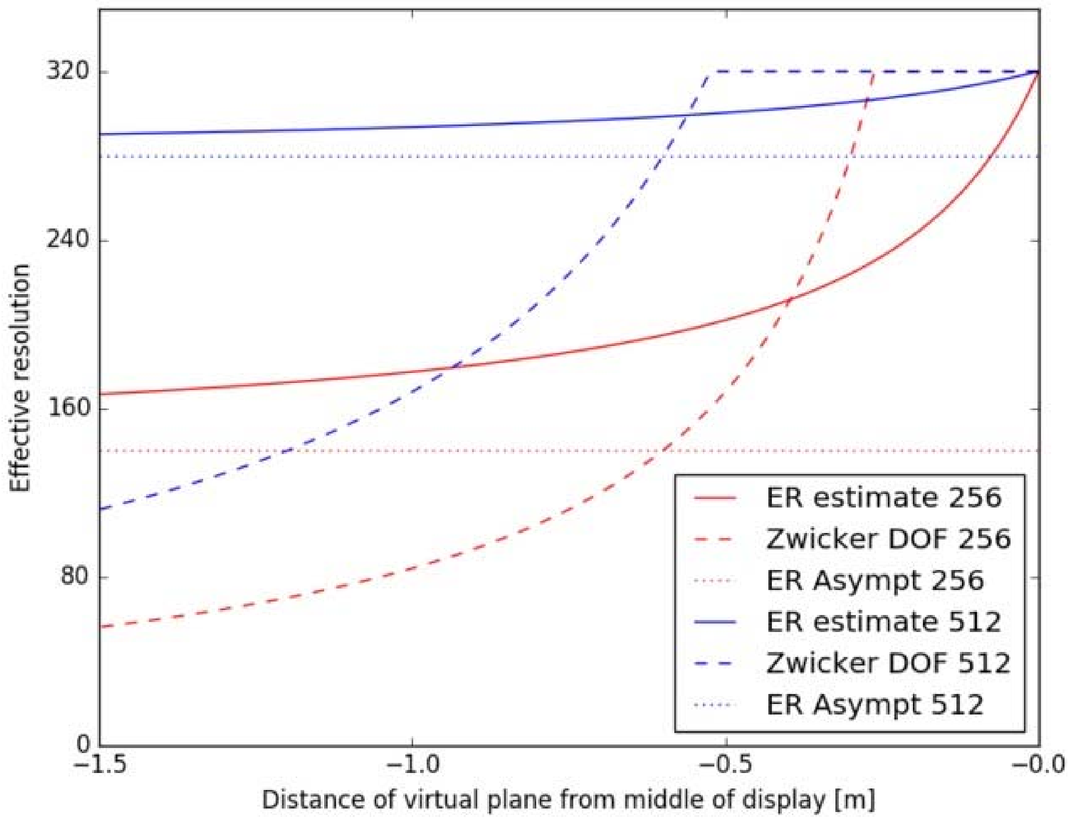

Figure D.4. Theory of Asymptotic Resolution at Depth

For the future, Hamilton believes we will need denser pixels on larger substrates (< 5 microns scaling to >24" panels). In microdisplays, pixels need to be in this same size range with larger substrates as well. He also sees advances necessary in high performing optics as well. Finer structures with tight tolerances will be needed for next-generation light field displays, he predicts. He pointed to promising work on-going at Harvard (Cappasso) and Caltech (Faraon).

Hamilton then mentioned some work his company is doing to model depth of field in light field displays. He thinks current models do not accurately portray how the resolution falls off with distance from the image plane. His model predicts an asymptotic roll off that starts sooner but levels off at a higher resolution than current models.

Finally, Hamilton mentioned the huge bandwidth challenge of light field displays. His company is working on a data compression scheme to address this issue, although he is not yet able to disclose many details.

D-1.3 Pete Lude, CTO of Mission Rock Digital. "An Overview of Light Field Acquisition"

Pete Lude, the CTO of Mission Rock Digital, was asked to provide an overview of light field acquisition in the entertainment arena.[69] He began his presentation with a history lesson in how far back the idea of a light field really goes. In 1996, Stanford University was playing with arrays of cameras, for example. He then cited patents dating back to 1968 (with a CRT); 1936 (defining the term light field), 1908 (defining integral photography), to 1846 (Faraday's "thoughts on wave vibration"), to 1492 (DaVinci describes radiant pyramids).

Next was a description of plenoptic light field vs. 2D and stereoscopic photography. Plenoptic light field capture features a main lens, as with traditional photography, plus a micro lens array in front of a main sensor (or series of sensors). See Figure D.5.

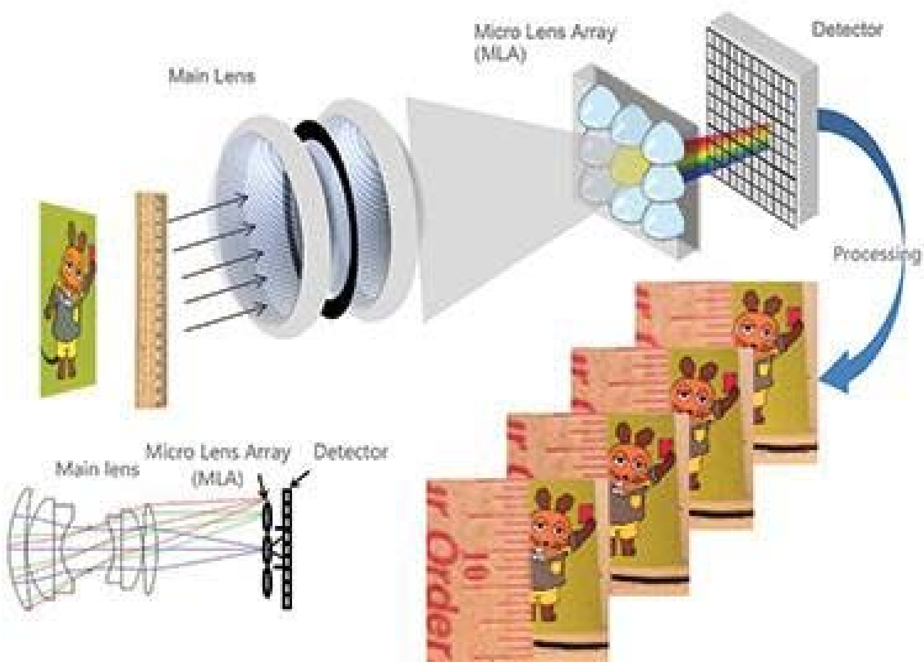

Figure D.5. Illustration of Plenoptic Camera Hardware

He touched on the storage requirements – they are huge: 500B rays * 3 bytes/ray for 8-bit color * 60 fps = 400 petabytes /hour.

Much of the focus today is on VR capture using a Big Ball of Cameras (BBOC). But these do not capturing a light field image – a stitched 360-degree image is a 2D or stereo 3D image only. One of the more impressive light field capture devices is the Lytro IMMERGE which is being tested now.

Light field capture systems are not only good for VR and entertainment applications, but they can work very well for automated optical inspection tasks as well. He also mentioned plant phenotyping, particle image velocimetry, and 3D microscopy as potential application areas.

The value of light field capture has been well documented by Fraunhoffer IIS and Lytro who have both done short films using the capture technique to develop the tools and show the benefits of the technology. With these tools, a high resolution 3D model of the scene can be built in a game engine with the video textures imported so it looks like the actual video. But since it is in a game engine, one can do virtual green screening to change a background, introduce new lighting or reflections, change focus and depth of field, do virtual camera movements, insertion of computer generated objects and more. Many think this is the future of cinematic capture as it allows so much flexibility in a post-production process. Fraunhofer IIS has packaged these innovations into a post-production tool set called Realception which is available today.

Lude then described other light field capture systems that use arrays of cameras (Microsoft) or camera on moving arms (OTOY) and others. His conclusion: light field image enables revolutionary imaging attributes, but the data is very large so new compression standards are needed. See Figure D.6 for an example a free viewpoint video streaming system under development at Microsoft.

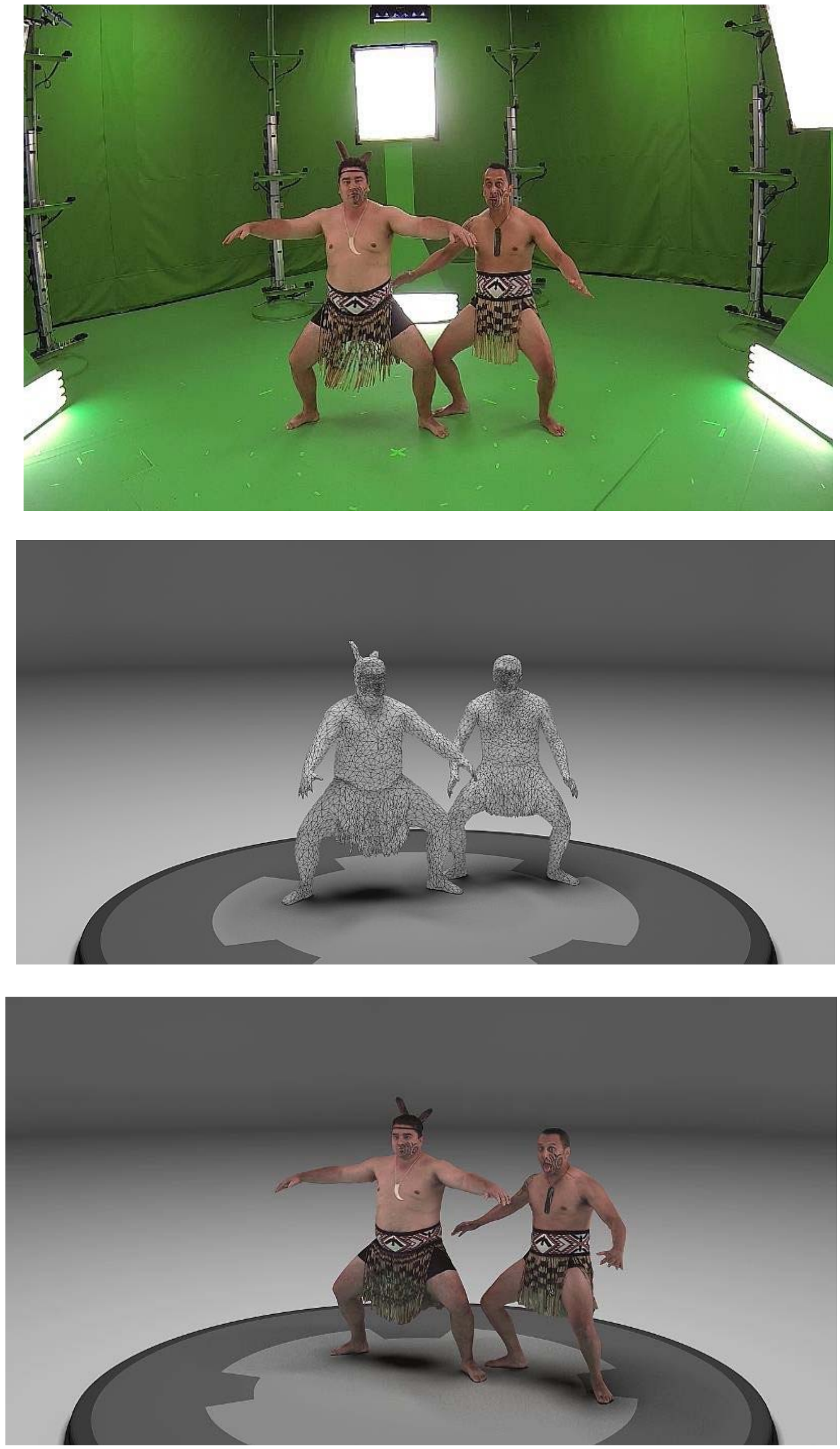

Figure D.6. Example of a Free Viewpoint Streaming Video Setup from Microsoft

D-1.4 Sam Robertson, Holochip Corporation, "Trade-offs in Light Field Streaming, Processing, and Display Requirements for High and Low Fidelity Applications"

Sam Robertson started his talk showing advanced displays from popular movies like Avatar, Prometheus and Iron Man 2 as well as video games like Hal 3 – Bungie.[70] He noted that many of these displays were more volumetric like with no occlusions, shadows or specular highlights. What are these good for, he asked?

How about architectural review? Not necessary for the rendering of the outside of the building, but yes for seeing how a room is located within a building, for example. The same can be said for command and control applications where seeing the whole airspace with assets highlighted, but transparent, might be very acceptable.

How about medical applications? Is it better to see a solid heart or spine or brain scan or a translucent version? The answer depends on the information trying to be derived or communicated.

His point was that not all advanced display applications should be treated the same. Some applications require high fidelity with bigger overhead while other applications can get by with lower fidelity. See Figure D.7 for a simulated display that would be very high fidelity and high bandwidth!

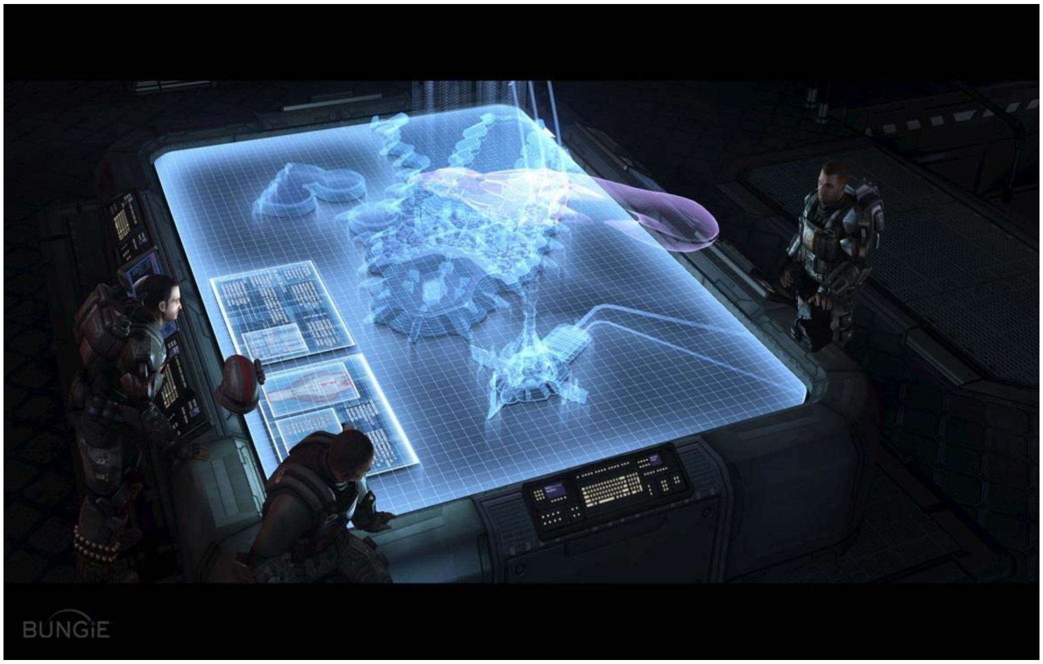

Figure D.7. Simulated Holotable Display from the movie Avatar

Low fidelity displays, such as the 4-view light field display by Leia that is in the RED Fusion cell phone, is an example. This will have a limited FOV and lower computational requirements, but the visual cues will be more limited as well.

Robertson said volumetric displays are in a category of their own as they offer accommodation and vergence cue, but no shadows, specularity or occlusion.

When developing a standard it is critical to consider not only the myriad of different display technologies which will all exist simultaneously in the market, but also that each of these need vastly differing amounts of data according to Robertson.

While Zebra imaging was creating holograms using data shipped to them on hard drives, the industry needs a solution which enables data to be streamed to a smart phone without much more bandwidth than required to show a movie from Netflix, he continued. He concluded by lumping advanced displays into the three categories shown below.

- High fidelity = all data, lots of processing (need an image for each)
- Low fidelity = all data, little processing
- Volumetric = less data, less processing

D-1.5 Nilo Maniquis, Naval Sea Systems Command, PEO IWS, "Improving Battlespace Awareness, Reducing Warfighter Workload, and Enabling Rapid Response Through the Use of Collaborative 3D Holographic Display"

Nilo Maniquis is from the Naval Sea Systems Command and he came to talk about what the Navy would like to have for advanced displays – at least for the DDG 51 Arleigh Burke class destroyer that he has been assigned to upgrade.[71]

Maniquis then described a very complex battlespace environment that is managed today in a rather crude way by young sailors. The battlespace is getting more and more complex increasing the warfighter workload. Decisions need to be made in a rapid fashion and mistakes can be catastrophic.

Having a 3D display where warfighters can see the situational battlespace with deep clarity, as depicted in the graphic below (Figure D.8), is the vision of the future.

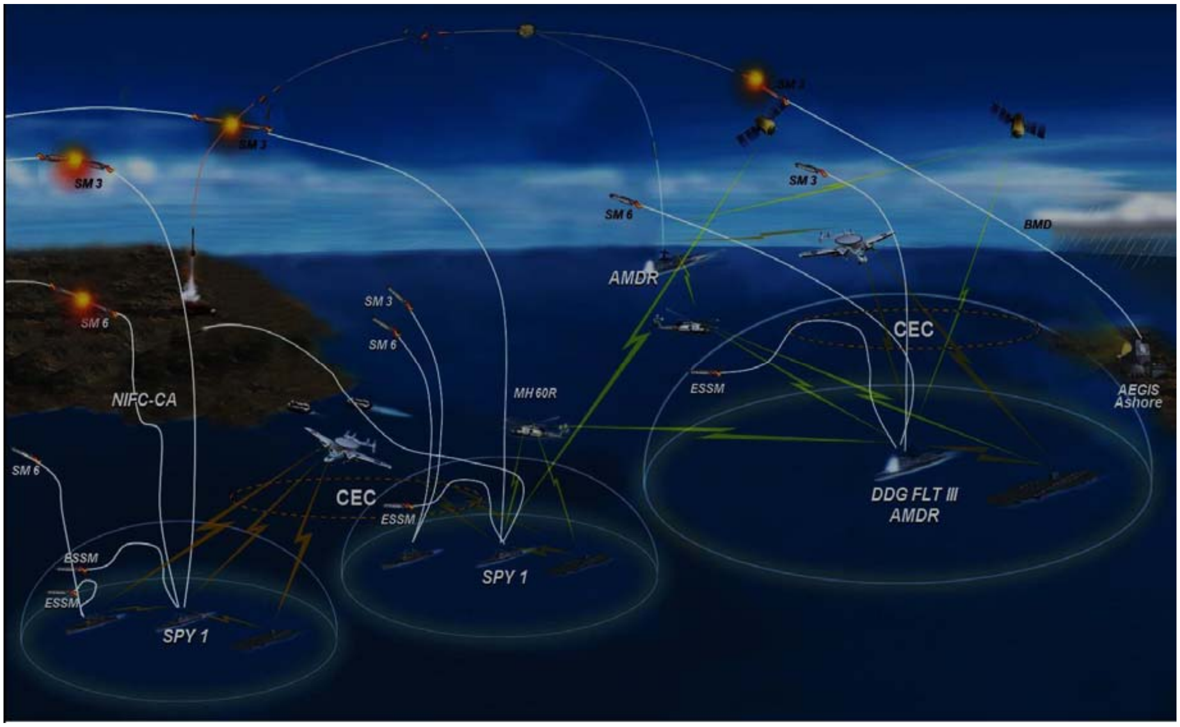

Figure D.8. The Future of Naval CIC (Combat Information Center) Displays

What sailors have today is shown below in Figure D.9. The lines show the direction of the assets with the length of the line indicating speed. This is not that much different from information available to warfighters in WWII! NOT SHOWN are key attributes like:

- Altitude
- Changes in altitude (increasing or decreasing)
- Type of aircraft or ship
- Range of sensors and weapons on each asset
- Link connectivity
- Track attributes and track history
- Real time systems status
- Platform point of view

Figure D.9. Present Day View of a CID Display. Items Must be Selected to Show Data

Just being able to selectively show this type of data would be a huge help in understanding the flow of the battle and making critical decisions. Having it shown in a tabletop collaborative environment is where the Navy wants to go.

Maniquis said that such display systems, and their associated software interfaces, need to provide a more intuitive view of the battle, which will lead to improved human performance. Understanding the track history (where it came from, maneuvers it has made, etc.) of an asset or a threat for example, provides a lot of insight into what that asset plans to do. That data must be remembered by the warfighter now.

Warfighters in the command center may also want to change their point of view – to see what a pilot or ship captain sees from their position.

In addition to the above data, overlays of additional information like sea lanes, air lanes, weather, multi-spectral images and topography will further aid cognition. 3D provides the warfighter a real time visual awareness and assessment of sensor, weapon, ship's systems and tactical/strategic performance, said Maniquis.

He concluded by saying that today's warfighter is still using 2D display technology from the 1970s in a time where everything is faster and more lethal. To fight better, we have to win the data information war. 3D displays will enhance the warfighter's ability to comprehend the situation and to make faster and more meaningful decisions.

Maniquis is backing up these desires by funding several development efforts to move in this direction. Figure D.10 is an example of the type of 3D display that the Navy needs to provide operators with full situational awareness of the battlefield.

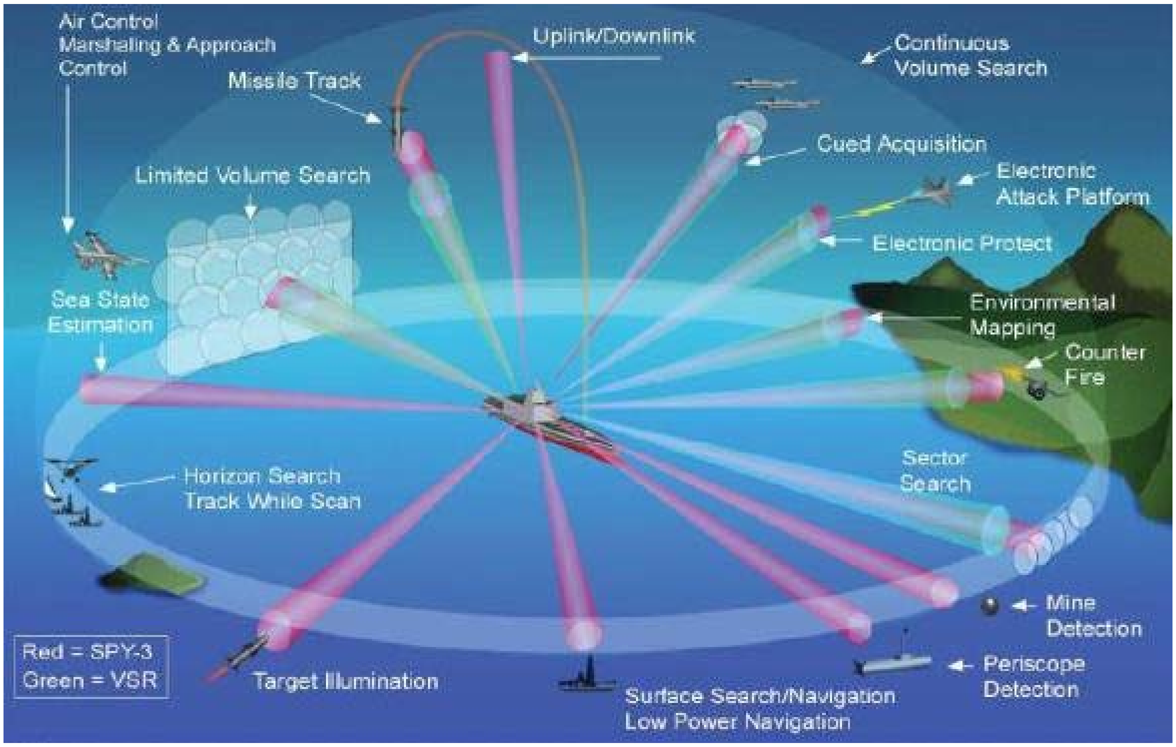

Figure D.10. 3D Real Time Visual Ship and/or System Performance for Operator Full Situational Awareness

D-1.6 Jamison Daniel, Oak Ridge Leadership Computing Facility, Oak Ridge National Lab, "Visualization Technologies and the Challenge of High Performance Computing"

Jamison Daniel from the Leadership Computing Facility at Oak Ridge National Laboratories spoke next.[72] He explained that the Oak Ridge Leadership Computing Facility (OLCF) was established in 2004 with Congressional funding to provide scientific discovery for missions that would have little commercial payoff and would be too costly for institutes or commercial enterprises. The OLCF operates a supercomputer called Titan, which is the nation's largest supercomputer for open science research (and #5 in the world). Most importantly, research at OLCF cannot compete with existing private business.

The basic capabilities of the Titan supercomputer are shown below:

- 27 petaflops of computer cycles:
  - Unique Gemini system interconnect (very fast network).

- 18,688 nodes (nodes can exchange data quickly using the Gemini network).
- 32 petabyte file system that can deliver 1 terabyte per second performance (record simulation results for post-analysis).

Daniel then switched gears to talk more about the visualization tools and techniques. He noted there are two main types of visualization needs:

- Analysis – where the answer is not known and the visualization is used to explore the data volume
- Communication – where the answer is known but a better way to communicate the results or impact are needed

Currently, OLCF has two visualization solutions shown in Figure D.11. The first is a flat videowall using DLP rear projection cubes (from Planar) and the second is a curved LED videowall (from Barco) with a resolution of 11,520 x 3240 (6 x 3 cabinets with FHD resolution). When a person is located 11.35 feet from the center of the curved video wall, their field of view is 120 degrees. But they have to be a bit closer, 7.45 feet for each pixel to subtend 1 arcminute.

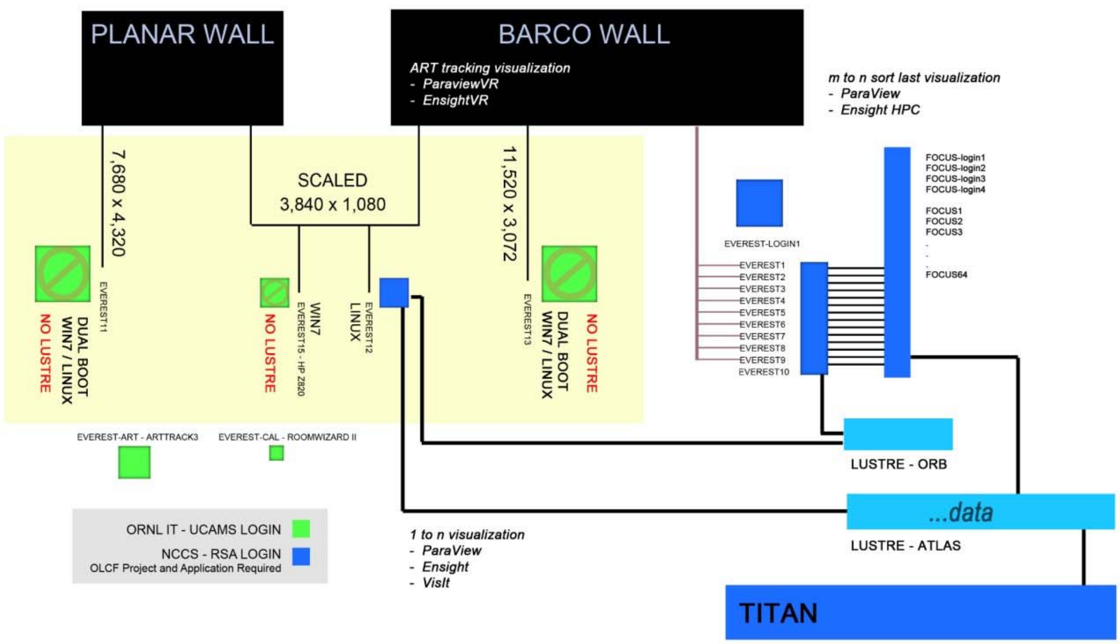

Figure D.11. Videowall Visualization Solutions at ORNL and a Diagram of Their Connectivity to the Titan Supercomputer

The curved system can support display of stereoscopic 11,520 x 3,240 resolution image at 48 fps using a data rate of 5,375 Mb/s.

While these capabilities are impressive, Daniel says they are increasingly looking at ways to offer remote visualization as it is not always convenient for their customers to come to the facility. This includes AR as well as light field display solutions.

Daniel also described a new architecture designed to work on particle simulations. This is based on 8 Tesla P100 GPUs (Graphics Processing Units) from nVidia for a node – with 16 nodes envisioned for the cluster. Apparently, this has been proposed but not yet funded by Congress.

D-1.7 JPEG-PLEN0, Walt Husak, "JPEG-PLENO's Interest in Light Field Images"

Walt Husak, who works for Dolby Labs, is also a member of the JPEG-PLENO standardization activity and the US national body representative.[73] He started his talk with an overview of how JPEG and MPEG are related to the ITU, ISO and IEC, as shown in Figure D.12 below.

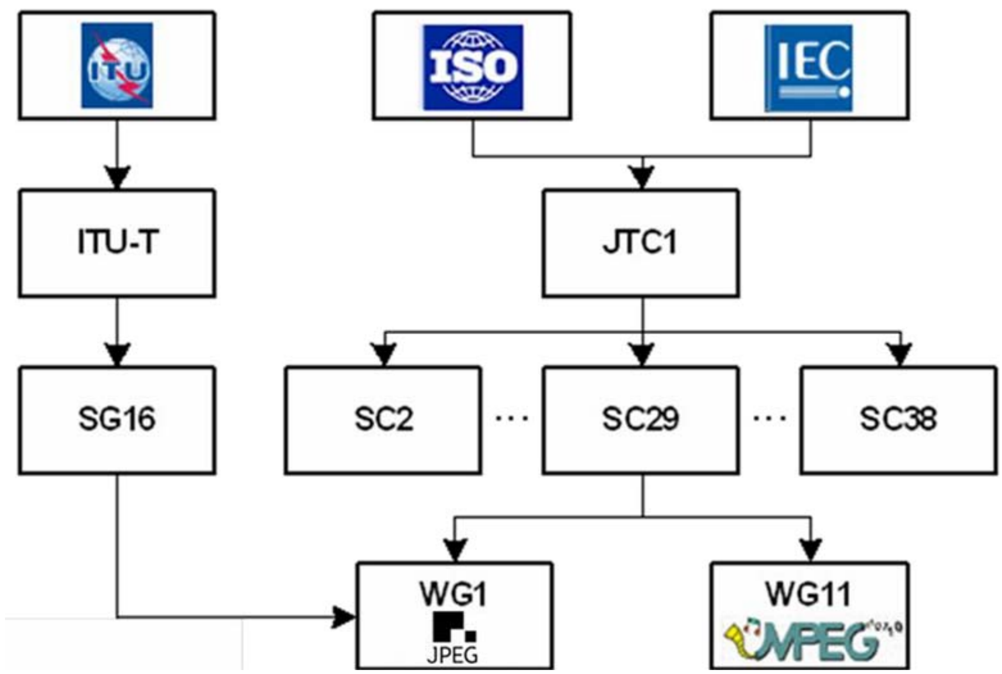

Figure D.12. Relationships Between JPEG and International Standards Organizations

The first JPEG standard, JPEG-1, was developed for still images and is widely used even today. For the cinema, medical and geospatial industries, it developed the JPEG2000 codec which compresses a series of static images in the movie.

The JPEG XT standard is an extension for JPEG-1 and is backward compatible. It is aimed at HDR photography applications and is structured to create an SDR layer and an HDR layer. The HDR layer is ignored by an SDR display.

JPEG XS is a mezzanine compression codec that offers light compression in the 2:1 to 6:1 range. This is currently in development. Future work is planned on a high throughput JPEG 2000 codec, a next generation image codec and a JPEG 360 codec, which will be useful for VR/AR environments.

Also in development is the JPEG-PLENO standard. Its focus is on compression of multi-view video assets for delivery to light field displays, VR/AR headsets and other advanced display devices.

So far, the group has had three calls for proposals for a light field coding technology and it is working on core experiments to be used to evaluate the usefulness of each proposal. Husak said the group will soon move from the competitive phase to the collaborative phase where the group will seek to take the best parts of each proposal to shape the final standard. The core experiments are designed to attack the candidate codec and to improve it.

The current schedule calls for the first working draft of the standard and the core experiments to be done in October, 2017, with the second draft by January, 2018. A committee draft standard will be issued in April, 2018 followed by publication in January, 2019.

JPEG and MPEG operate via ad hoc groups that form for particular tasks for a 3 month period. For example, the coding and analysis subgroup within JPEG PLENO has a mandate to run core experiments as defined during the 76^{th} JPEG meeting; to cross check results and design software interfaces and modules for the next set of core experiments. This should all be done by the next JPEG meeting on Oct. 21-22 in Macau, China. Such ad hoc workgroups are mainly staffed by institutes and universities.

Husak then showed the generic JPEG PLENO architecture (see Figure D.13) and discussed the details of the current core experiments that will be used to stress each candidate codec.

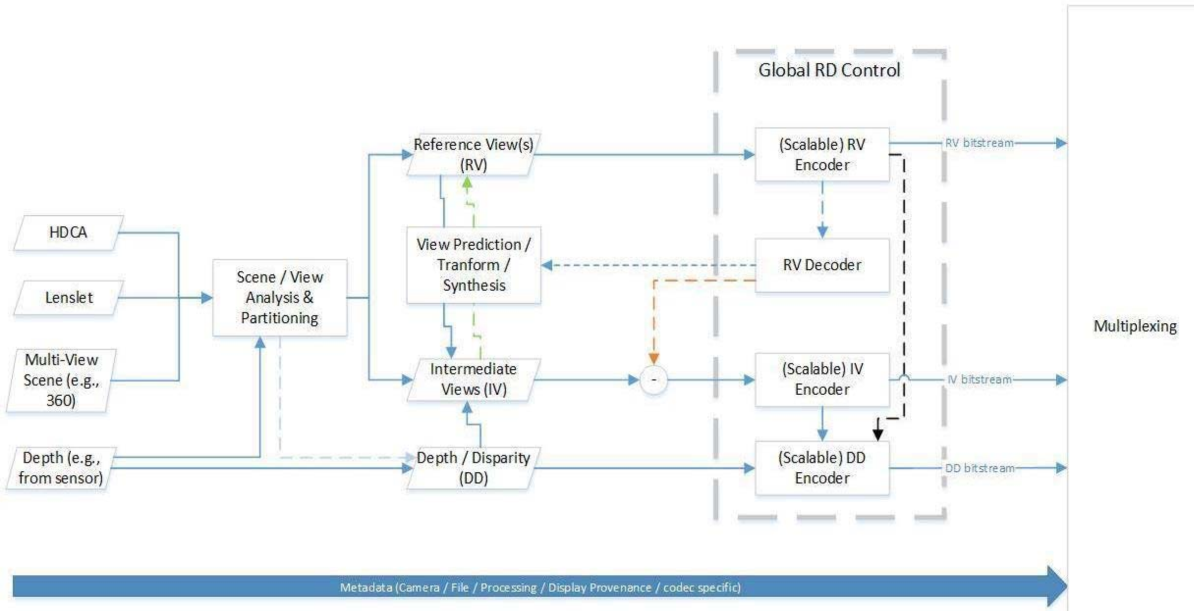

Figure D.13. JPEG-PLENO Architecture

- CE1 Scene/View Analysis & Partitioning
- CE2 View Prediction, Transform & Synthesis
- CE3 Depth/Disparity Representation & Coding
- CE4 Texture Coding (RV + IV) □ Postponed until next meeting

While each core experiment has different goals, test data and evaluation criteria, PSNR (Peak Signal to Noise Ratio) and SSIM (Structural Similarity) are often used along with other parameters. Husak concluded by noting that JPEG continues to look for effective criteria for subjective test methods.

D-1.8 C. E. (Tommy) Thomas, Third Dimension Technologies, "SMFoLD Streaming 3D Media"

Moving to the streaming proposals part of the program, Tommy Thomas of Third Dimension Technologies (TDT) kicked off this section by summarizing the goals and objectives of the Phase II award, the contract partners and the needs driving the development of a light field streaming standard.[74]

Thomas explained that light field displays need to render 3D scenes from multiple viewpoints, but the 3D applications don't provide all the needed information and information in precompiled shaders is inaccessible. Plus, there are many data types and graphics rendering interfaces so a new standard is needed.

TDT's approach is based upon using a graphics representation of the light field data and OpenGL protocols. This will limit the structure of the light field data to mesh, texture and OpenGL primitives. With their approach, precompiled shaders with named variables are used as these are oriented toward a single viewport application, rather than standard shaders or precompiled shaders. Instead, they will create a series of named variables that will be used in the precompiled vertex shaders. Variables will cover aspects such as camera position, camera angle, focal plane, camera field of view and others.

Each source application will have to link to and SMFoLD DLL and use named variable metadata in its shaders. Each display device will have a new SMFoLD DLL as well to accept the names variables and render to desired views. A simplified flow diagram is shown in Figure D.14 below.

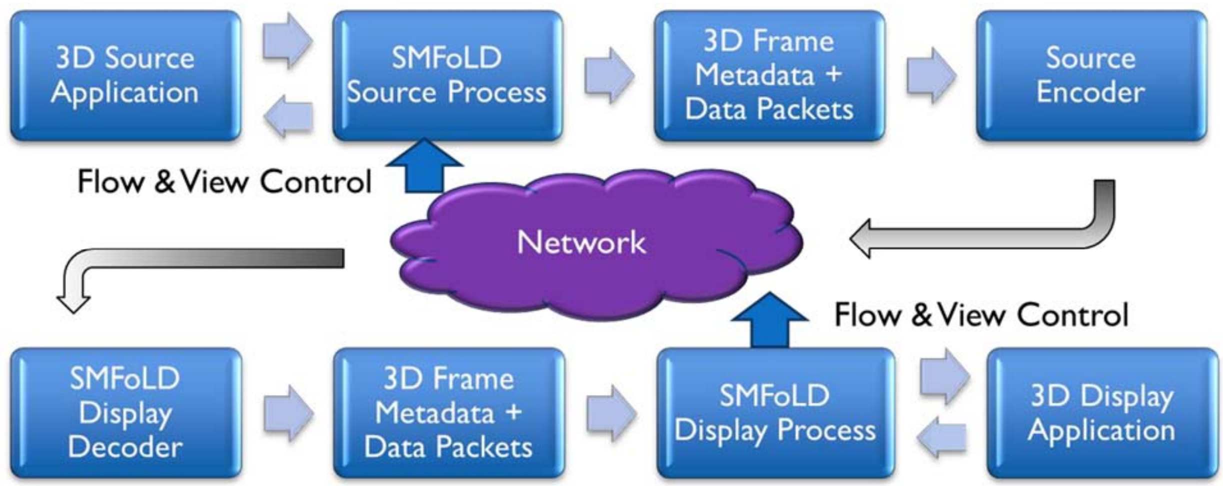

Figure D.14. Nominal SMFoLD Work Diagram from Source Application to Display Application

TDT has also coined the term "3D Frame." This is supposed to include all the information needed to display an image on a 2D or 3D display. It must include values of the function calls, data that the function calls use and metadata to allow the display to create any number of viewpoints. It also includes geometry transformation matrix, colors, material properties and other texture data, plus arrays of values expressing 3D structures or models. Shaders provide the rendering pipeline logic for all frames.

Like in a game engine, object models can be downloaded and stored locally allowing local manipulation of the objects without changing the object data.

Thomas then showed some typical data rates for popular applications like Google Earth with high data rates should this be distributed as light field data. He then listed some existing compression and encryption approaches the team plans to evaluate during the contract period for use with the proposed standard.

More details of the proposed standard were discussed including irregular frame rate issues, audio, backwards compatibility and more. Thomas said that during the Phase II project, TDT and partners intend to better define the elements of the proposed standard and then demonstrate and end-to-end solution from source to display.

D-1.9 Thomas Burnett, FoVI3D, "Object Graphics Library (ObjGL) and the Heterogeneous Display Environment"

Thomas Burnett from FoVI3D then presented their approach to a light field streaming model. He started off by focusing on a local light field display system based on a mesh and texture representation where the data is already stored on the server.[75]

In a typical graphics pipeline, data is passed from the server to the client processor system where a 3D application creates a scene and provides functionality. The render engine then describes the scene through a set of draw commands that are passed to the GPU for rendering into a video stream that can be played back on the 2D or stereoscopic 3D display. See Figure D.15.

The problem is that if the display architecture changes, i/e. becomes a light field, volumetric or holographic display, the 3D application needs to change as well. In addition, this current architecture is designed to render a single 2D view for a 2D monitor. Stereoscopic 3D can be supported, but requires two sequential render passes, slowing down the frame rate.

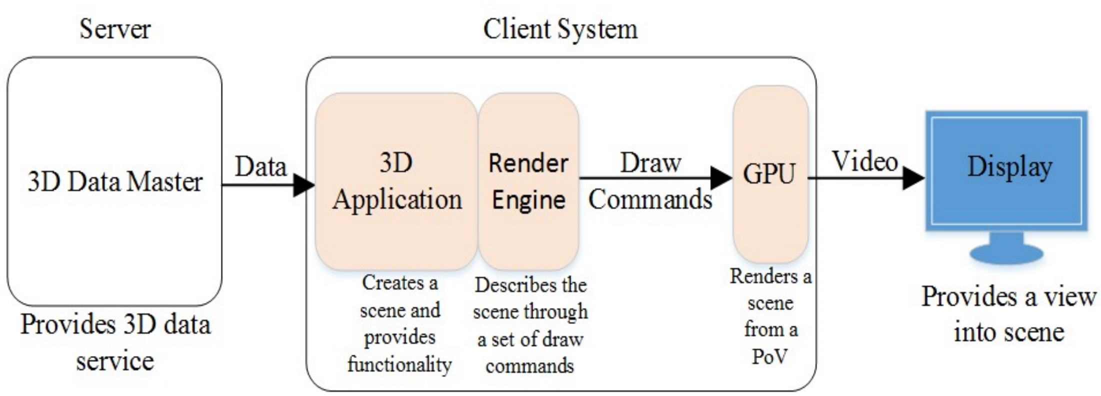

Figure D.15. Typical Client/Server System for Rendering to a 3D Display

To address this problem, FoVI3D is proposing new extensions they call ObjGL. This adds control where the draw commands are being generated and potentially allows any type of display to be used and for the source data to now be located remotely as well.

Burnett then went into more detail on some of the problems with OpenGL and GPUs when used for multi-view rendering, which is needed for light field and volumetric displays. He explained that an "OpenGL Shim" is needed with graphics-based rendering on multi-view displays to “hijack the command stream unbeknownst to the host application.” This, in combination with a multi-view display-specific library of commands, can be used to render the multiple views.

While this works, there are a number of issues which he highlighted. For example, the shim/interceptor approach does not work as well with the newer OpenGL architecture that moves away from the fixed pipeline to a more programmable one. It is hard to draw line and points and many of the commands are just not constructed for multi-view rendering. As noted earlier, OpenGL was designed for a writing a single view to a frame buffer and there is no agreed upon way to render multiple views. This can lead to a stall in the host application as sequential views are rendered. This ends up pushing multi-view render responsibility back to the host application – which is not a good architecture.

As TDT noted in their presentation, trying to send light field data as pixels requires huge bandwidth and means the scene is rendered with the source point of view. As the user moves around the scene, he may see “holes” that are missing pixels that were occluded by the source rendering.

As a result, FoVI3D and TDT both recommend delivering light field data as mesh and textures (polygons/model data). This means sending the geometry of objects ahead of time for pre caching and then sending transforms to manipulate the objects. These transforms are not too large – a 4x4 matrix and a transform command would be on the order of only 68 bytes.

Shown below in Figure D.16 is an alternative 3D rendering architecture where each view of a display has its own rendering engine.

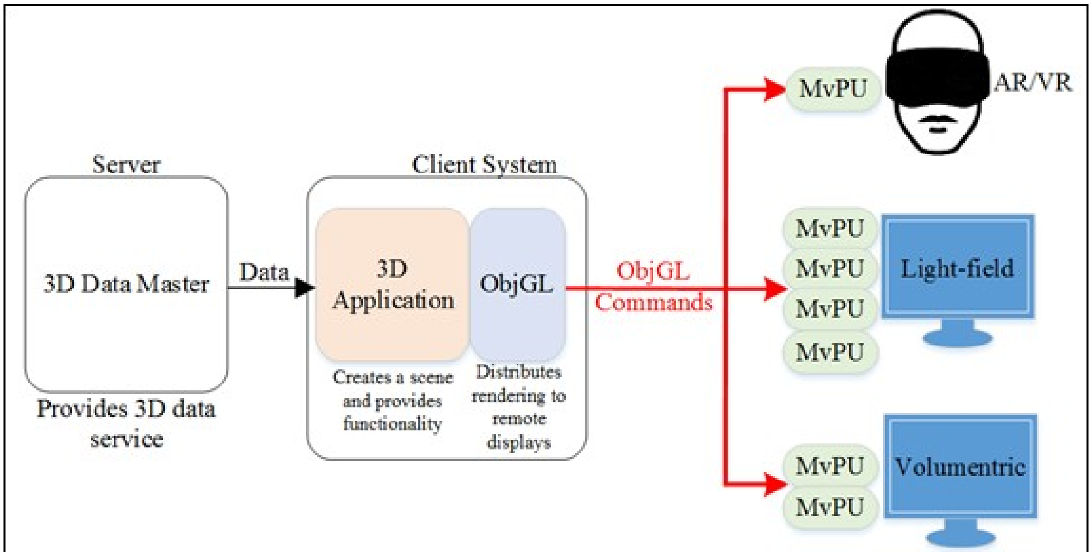

Figure D.16. Alternative 3D Rendering Architecture With a Rendering (MvPU) Engine for Each View of a Display

FoVI3D's approach is to develop an Object Graphics Library (ObjGL) as a replacement for OpenGL. It supports a display agnostic requirement and multi-view rendering and separates the host application from the display. The approach can support traditional first person rendering for 2D or stereo 3D displays, but can also support other advanced multi-view display architectures. They call this the display centric perspective which defines a position and orientation of a view volume. The scene is then rendered from the perspective of the display instead of the viewer.

FoVI3D has also proposed a new GPU architecture they call Multi-view Processing Units (MvPU). This essentially takes a serial process and makes it a parallel one, locating these MvPUs in the display, not in the client system (i.e. PC).

D-1.10 Arianne Hinds, CableLabs, "Toward the Realization of 6 DoF Standards"

Arianne Hinds works for CableLabs, the technology development group for the cable industry, but she was at the workshop to talk about activities in MPEG-I. Like Walt Husak, she is the national body representative for MPEG in the US.[76]

Much of the activity in the MPEG-I group today is focused on the needs of VR. As result, she first started out by talking about the 6 Degree of Freedom (DoF) workflow being facilitated by a media container known as ORBX. ORBX is a container format developed by OTOY and other 3D modeling and rendering companies that "supports a minimum set of interchange formats to specify a scene graph." This is a high end container that integrates many of the assets and tools used by visual effects artists, graphics artists and post-production professionals. Over 30 tools already support the format, so it has industry support.

OTOY has taken the ORBX format to MPEG-I to see about getting it standardized. CableLabs is helping to make this happen.

Hinds then described how Facebook is using ORBX in its VR 6 DoF workflow. First, Facebook now has two VR camera rigs; one with 6 cameras and one with 24 cameras. This raw video content is placed in the ORBX container and uploaded to the cloud. There, the video images are analyzed to create a 3D model of the scene described with meshes, textures, point clouds and more. ORBX contains named variables in the associated metadata to drive the shaders for rendering. This allows the scene graph to be display agnostic meaning it can playout on an Oculus Rift, and HTC Vive or other headset without having to have a separate version of the content for each headset (as is needed today with every other solution).

Hinds even went so far as to say that CableLabs is considering adopting this workflow for the cable industry as an alternative to delivering compressed video. That would be a huge industry change.

But there are still some missing pieces to allow interchange/deployment over networks. For one, there needs to be a standard interface from the scene graph and media container to the render engine. There have to be agreed-upon primitive formats (i.e. a minimal set of meshes, point clouds and textures) and there must be photorealistic rendering. Also missing is a good method for compression of meshes, point clouds and floating-point/high bit-depth textures. And, you need an infrastructure to support distribution.

OTOY has a sophisticated ORBX render engine that can create photo realistic images that look incredible (and are used today on high profile Hollywood content). She then showed some examples rendered from light field source data.

ORBX is based on the Open EXR format initially developed by Industrial Light and Magic. It supports 16-bit floating point and 32-bit integer formats and an arbitrary number of channels. It can support point cloud simulation for fog, smoke, fire and clouds, such as the OpenVDB format initially developed by Dreamworks (see Figure D.17. Mesh support comes in the form of Alembic, a format developed by ImageWorks that provides complex computational and procedural constructions to render hair, grass and other fine textures.

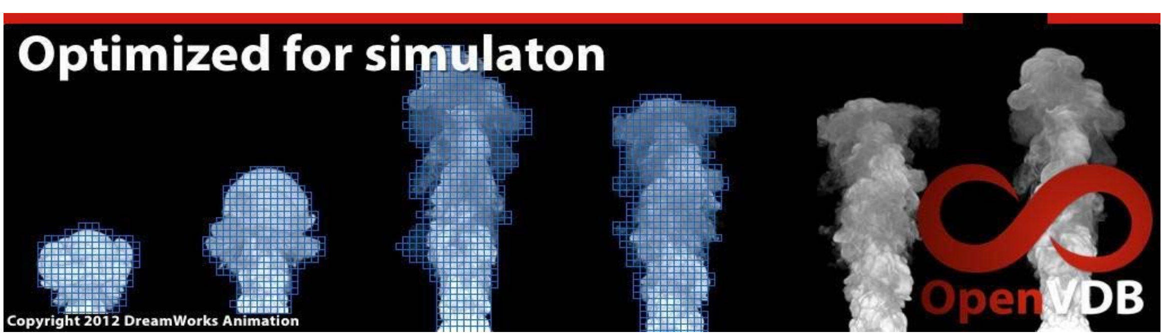

Figure D.17. Examples of Mesh and Texture Rendering Using Open VDB and Alembic

And there is more. ORBX also support fundamental attributes for velocity of objects, motion, light sources, etc. and it can support compressed formats as well.

Hinds then turned her attention to activities in the MPEG-I workgroup. She said the work on 3DoF standards are done or nearly done, so most activity is on the 6 DoF standards. Here she listed the following activities:

- Light field compression experiments
- Point cloud compression (CfP)
- Hybrid Natural/Synthetic Scene
  - Container for 6 DoF assets
  - Scene graph
  - Identification of minimum set of formats for interchange (e.g. EXR, OpenVDB, and Alembic)
  - OTOY ORBX submitted as a candidate technology (Type-1 licensing)
  - Plan is to issue CfP
- 6 DoF Audio

She concluded by noting that ORBX will be royalty free (Type 1). She anticipates a final standard being issued in 2020 or 2021 and requested that TDT start working with MPEG now to get their proposal in the pipeline for consideration.

D-1.11 Jon Karafin, Light Field Lab, "Holographic Content Considerations"

Jon Karafin from Light Field Lab was the last presenter. He started out by trying to dispel some inaccuracies around light fields.[77] For example, a stitched 360 VR image is not a light field image (just 2D or stereo 3D). Point cloud data is not a light field – just a 3D volume of points of data. A "deep image" and textures are not light fields either. The "Tupac hologram" is not a light field (it is a Pepper's Ghost configuration). Princess Leia is not feasible either as you can't freeze light in mid-air.

Karafin said there are three key elements to consider in creating a compelling light field or holographic display.

- Rays per degree – higher ray density is better
- View volume – controls the amount of freedom the viewer has to move in a given light field space
- 2D equivalent resolution – determined by the number of active rays that can be delivered in any 2D slice in space

There is also a minimal dataset to create a light field volume, described as:

- RGB (Red, Green, Blue) data for array samples – these viewpoints help define sample density, viewing volume and overall volumetric quality

- Surface coordinates per sample – accuracy is essential for these coordinates, otherwise temporal and spatial artifacts will result
- Virtual-camera coordinates and metadata - maintaining a singular world coordinate system is key to aligning the light field projection to the interactive camera coordinates

Turning to the bandwidth issues with light field display, Karafin said that transmitting uncompressed raw light field data is not possible as this requires more than 5000 GBps (GigaBytes per second). In what he called hybrid processing, the light field data can be "vectorized" into a format that can reduce bandwidth needs to perhaps 300 Mbps. The third option he sees is volumetric representation that can be streamed in the 10-30 Mbps range.

Karafin favors the hybrid processing approach and then showed the following flow diagram (Figure D.18). Light field data can be generated in several formats, including live action. If fully synthetic, this can be streamed directly to an end user. Otherwise, content is converted into hybrid data, which was not well defined, but probably means the conversion of live action and other content into a model/textures format. These can be very complex models and associated extensions as he proposes using the ORBX container (see capabilities of this from the presentation given by Arianne Hinds). Karafin says this approach allows for the retention of key light field qualities like refraction, reflection and transmissivity of objects.

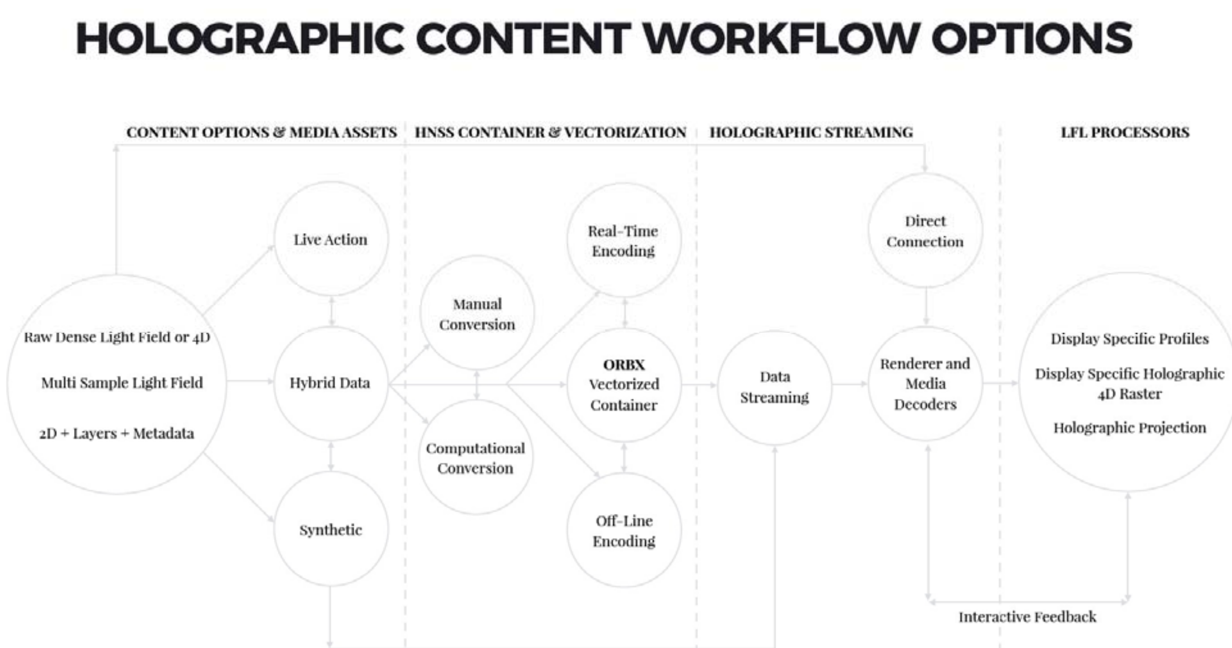

Figure D.18. Flow Diagrams for Light Field Streaming

Note that this approach separates the application from the display rendering so the application won't stall based upon display performance (one key need pointed out in the FoVI3D presentation). And, a 2D layer can be created as well than can be streamed in the 30 Mbps range for conventional displays.

D-1.12 Panel Discussion

At the end of the day, all of the speakers were invited to come to the front of the room (Figure D.19) for a panel discussion with moderator and audience Q&A.[78]

Moderator Chris Chinnock started the session by showing a potential list of SMFoLD evaluation criteria that was quite extensive running for 3 slides, as shown below:

- Is it an open standard requiring no licensing fees?
- How difficult is it to implement in applications? …on displays?
- Are tools and reference models available to support implementation?
- How difficult to port existing applications?
- How steep is the learning curve?
- Is it truly display agnostic?
- Does it allow the display application to control flow and view point?
- What is transmission bandwidth requirement?
- How well does it perform? .. latency? ... image degradation?
- How well does audio sync with visuals?
- What data types can be streamed?
- How likely to be adopted?
- Does it provide for encryption?
- What is the minimum acceptable streaming rate in FPS (this is potentially quite variable in Mbytes/sec)?
- Is it acceptable not to necessarily support all legacy applications, or to require the source end to provide manual support of some parameters (potentially set and forget)?
- What is the minimum acceptable level of OpenGL support? Is OpenGL 2.1 an acceptable starting point?

He asked if there were any comments on these from an end-user point of view. Arianne Hinds said they were looking carefully at application support and interchange. ORBX is already supported by 30 tools and OpenEXR is an open standard too, so a great starting point.

Pete Lude wondered how one might evaluate image quality in a light field display. There are many items that can degrade the image so we need new metrology to address this.

Thomas Burnett said that they are working on LF metrology, but he said that the added value of a 3D image can in some ways outweigh the lower resolution compared to a 2D display. It is the 3D fidelity that is the key to evaluating a light field display – the metrology for which needs development. Lude added that frame rate will be important too.

Figure D.19. Speakers During the Panel Discussion at the SMFoLD Workshop

Burnett emphasized that a 3D situational display is really important to see if a plane is climbing, for example. Nilo Maniquis stressed that immersion is really important to better decision making in a fast-paced environment. He said that the Air Force, Navy, Army and NGA are all putting funds into what he believes the mainstream challenges are. But there are not huge R&D dollars behind this yet. The DARPA Gen 1 was the first effort to look at advanced displays. Static holograms are useful as well for training too.

He then described a use case where the next command center for a ship for 2025. The design was built into CAD and he had a static hologram made from the CAD data for $300. In 10 minutes he saw three major flaws that would have cost tens of millions of dollars in correction. An audience question asked about connectivity. Karafin said that even for the 300 Mbps scenario of the hybrid solution he described previously, the data rates are manageable using existing solution. Moving raw data can't be moved today. While the GPUs can perform extremely well, the output is fundamentally limited by DisplayPort. So you can make the data fit into one or more DP connections.

Chinnock noted that there is a U-SDI standard that offers 24 ten Gbps connectors in one bundle.

Burnett stressed that the idea behind their MvPU proposal is to move this to the display so the application does not have to handle a big bandwidth. They send geometry and transform in the MvPU.

Tommy Thomas said that you only need HDMI for each projector in a horizontal parallax only solution. He then asked why you may need vertical parallax. He then asked if OpenGL frames can be packed up on ORBX.

Karafin though the format was agnostic as it packages the texture, but he thinks of ORBX more as a storage format than a streaming format. He likes the idea of being able to do this with OpenGL calls as it saves a lot of headaches, but this may not be the same as the ORBX approach. More discussion is needed.

Karafin then questioned if HPO really solves the vergence-accommodation problem saying this is only so if you have a "slit pupil." Thomas countered saying the eye collects in both directions and that the image is real as you can see rays converge in front of the screen if you place a piece of paper there. Karafin seemed surprised by this statement and said he would like to see this. Chinnock asked if we need to develop a glossary of terms as there seems to be some confusion about similar terms. Hinds said she thought this is necessary as they have already found it difficult to communicate even among these experts. Jon Karafin and Jules from OTOY come from a VFX background, MPEG experts are more 2D coding experts, plus optics people and others. A lot gets lost in translation. They have already had requests to add definitions for key terms.

Burnett explained that with their MvPU concept, you plug the display into the network and it natively renders the content. There is no external PC needed. He has taken this concept to NVIDIA, AMD and others and everyone is intrigued with this idea of reliving the PC of the rendering responsibility. Nobody who makes a commercial GPU has shot him down on this, which he finds very interesting, but he suspects they are realizing that the current approach of rendering a single view is not the way it seems to be evolving. With a variety of output devices that require different render pipelines (like for a Rift vs. a Vive), they may be realizing they have to support multiple views with a single geometry. He thinks of MvPU as hundreds of little GPU rendering chips right in the back of the display.

Hinds says she totally agrees with this architecture and that at CableLabs, they are looking at designs to put some of the rendering in the network to move some of the burden out of the home and into the network. It might get rendered to a point for example, with the rest done by the device without requiring a lot of compute power. Let's call this distributed rendering, she said.

An audience question asked about delivery possibilities by 2020 to support a consumer-level display. Hinds said they do have specs for more cable/network bandwidth than what is deployed, but until there is a reason to deploy, there is no economic incentive. The demand has to be there.

Thomas thinks the driver needs to be in the display for interactivity, but you essentially are building a PC in the device. You should start with the driver in the PC and then move to the device.

Chinnock asked if by 2021 it would be possible to realize the needs as outlined by Maniquis and Jamison Daniel. Responding to the question, Daniel noted that the data generated by a supercomputer is so massive it may not be reasonable to think that could be streamed to any device any time soon. The end product is likely to look more like conventional images after post processing. Is it supercomputing, visualization or a video game, he asked, saying it may not matter at this latter stage of the visualization workflow.

Thomas thinks it will be possible to stream the type of data Maniquis needs if horizontal parallax only. Full parallax will be much more difficult.

Matthew Hamilton thinks putting rendering at the display makes sense for some applications but lower fidelity applications can still use DisplayPort and shaders.

Sam Robertson said they are focused on finding the minimum requirements for the rendering. Does every application need full speculars, occlusion and shadows, for example, he asked, because if not, that simplifies the solution for short term applications.

Chinnock asked Walt Husak as the JPEG representative, Hinds as the MPEG representative and Lude as a SMPTE representative to give some advice to companies that want to develop a streaming standard for light field displays.

Hinds said that ad hoc groups are organized by MPEG and JPEG and there is a good description and ways to participate, so this is a good way to get started. You don’t have to be a member to be on these lists. These are open to the public.

Husak agreed with Hinds but said that some people view engagement as daunting. He and Hinds can help by having a sponsor to get an input document started. This is useful if you have a use case to look at and you want a standard to do something for you.

Lude said SMPTE is focused on the entertainment ecosystem. The standards are formed more on a systems basis once the core technologies are in place, for example, he said, that JPEG2000 was used for digital cinema. The 10e committee for essence is one group to look at and there is a quarterly summary of the various standards group activities on the website as well. There are also many engineering reports. He thinks it is now time for SMPTE to start looking at requirements documents for light fields in entertainment. Husak said that exact topic was raised in the meeting 2 weeks ago, so watch this space.

An audience question asked for some more details on ORBX to which Karafin said that the format supports animation and special effect – any format really. But you have to ask if the render engine can handle the format as well.

Burnett also provided some clarification saying that if you capture data from a singular point of view, then you will have occlusions in your image. The only way to fix this is to send geometry. For example, if you project a radiance image from a 2D Netflix movie, you will have occlusions where there are no pixels. You have to create the geometry from the video or capture it directly.

Karafin noted that ORBX is a data container whereas what Burnett is proposing is a data structuring method with OpenGL and ObjGL. ORBX can support the transfer of geometry data in a mesh and textures format indirectly.

Lude was asked how the entertainment industry is looking at light fields. He said that interest is starting with AR/VR and immersive cinema experiences, but this creates unique challenges because the director cannot control the view directly now. Every studio is looking into this to understand the impact on storytelling and he thinks the same thing will happen with light field acquisition and post production.

When it was pointed out that if a light field cinema existed, the experience would be different in every seat. Lude said this was not unlike going to an opera or a play, so the director must use different techniques like lighting and verbal cue to direct attention to where they want it. But, is a more lifelike cinematic experience useful, he asked? 24 fps movies have a lot of motion artifacts, but we are used this “film look” so is something more real better? Certainly for some genres more realism may be better.

Is the release of the new RED Fusion phone with a Leia 4-view light field display of interest to Hollywood, asked an attendee? No, was Lude's quick response.

Should Khronos be a group that SMFoLD should be engaging with? Yes, for anything geometry based, said Burnett. Hinds said they have certain standards organizations they turn to for content traveling over the cable network. OTOY submitted the ORBX spec to MPEG because this is one of the key standards organization the cable industry looks to for guidance. So she is not convinced Khronos is a group the cable industry would turn to.

In a follow up email, Hinds listed the SDOs (Standards Development Organizations) most active in the distribution part of the ecosystem as:

- For mobile distribution: the SDO is mostly 3GPP
- For cable distribution: SCTE, CableLabs, MPEG
- For satellite distribution: MPEG and maybe ATIS (I'm not sure)
- For WiFi: IEEE and WiFi Alliance
- For web: W3C

Chinnock asked how military and government users might leverage light field activities going on in entertainment and consumer electronics.

Husak noted that government is not profit driven. Money in this sector is flowing into technology, not VR programs.

What will a light field display offer in a simulator that will offer better training was asked by an attendee. Any benefit needs to be traced back to meeting a requirement that is not currently being met, he said. There are some applications where 3D is valuable like aerial refueling, helicopter landings, aircraft carrier landing and close terrain air support, for example.

D-1.13 Conclusions

1. There was considerable enthusiasm for a 3D streaming standard at the SMFoLD workshop.

2. There is not presently any general agreement on what that standard should look like.

3. Several standards bodies and organizations are working on SMFoLD. These include:
    a. Third Dimension Technologies is developing an OpenGL based 3D Streaming media standard which will require linking with an SMFoLD DLL and will also require named variables for metadata in the Source Application OpenGL shaders. The requirements on the Source and Display Applications and the network are very light (very light for the network by many orders of magnitude compared to a pixel stream), although all the heavy lifting (rendering of multiple views) must still be done by the Display Application at the Display using the OpenGL 3D frame and metadata supplied by the SMFoLD application.
    b. FoVI3D is working on a new API called Object Graphics Library (ObjGL) as a replacement for OpenGL. This separates the host application from the Display

and then the ObjGL command are sent for rendering to an MvPU processing unit(s) attached to the display.

c. MPEG-I is working on a 6 DOF standard using the ORBX container combined with applications that work with the container. This should work very well for cinema applications but may be more difficult to apply to real-time streaming applications.
d. JPEG-PLENO is working on light-field coding technology—codecs for compressing light-field data.

4. Field of Light Displays are coming, in one form or another, and there is a definite need for standards to deliver content from Source to Display. Multiple efforts are under way on standard development, and work from all of them may ultimately be combined to provide Streaming Media for 3D Field of Light Displays.

D-2 TDT SMFoLD Workshop 2 – 2 Oct 2018, Harman International, Northridge, CA

D-2.1 Introduction (Agenda, see Table D.2)

Table D.2. Workshop Agenda

Company	Contact	Abstract	Start
Insight Media	Chris Chinnock	Welcome	9:00 AM
Aclertic Systems	Yahya H. Mirza	Creating Content for Emerging Light Field Displays	9:15 AM
Google	Ryan Overbeck	The Making of Welcome to Light Fields	9:35 AM
CableLabs	Arianne Hinds	Update on Light Field Standards	10:05 AM
Ostendo	Zahir Alpaslan	JPEG Pleno: A Standard Framework for Representing Plenoptic Modalities	10:25 AM
Visby	Ryan Damm	Beyond Lumigraph – Parametric Light Fields for Capture and Display of real-world content	11:05 AM
Third Dimension Technologies	Tommy Thomas	Updates on Development of An Open Streaming Media Standard for Field of Light Displays (SMFoLD)	11:45 AM
Panel Discussion	Chris Chinnock		12:05 PM
Light Field Labs	Jon Karafin	Even Further Beyond VR: Light Field and Holographic Technology Updates	2:15 PM
Holochip	Samuel Robinson	Scalable, Real-Time Lightfield Rendering	2:55 PM
FoVI3D	Thomas Burnett	Enabling the Heterogeneous Display Environment	3:15 PM
LightSpace 3D	Ilmars Osmanis	Volumetric Displays for Sand Table and HMD Applications: An Update from Light Space 3D	3:55 PM
Panel Discussion	Pete Lude		4:35 PM

D-2.2 Yahya Mirza, Aclectic Systems, "<u>Creating Content for Emerging Light Field Displays</u>"

Yahya Mirza from Aclectic Systems began the conference with a discussion of the computational needs for rendering light field images. His company's goal is to "better understand what it takes to create content for future light field displays and look for opportunities to automate complicated and time- consuming processes so holographic productions can be more economically viable."

To get there, Mirza is developing plug-ins for special effects and compositing tool sets and working with Intel to build state of the art processors to support such rendering. It is still early in development, but he has a 9-layer approach to rendering synthetic light fields that can include very hard-to-render items such as smoke, transparency, glass, textured metal surfaces and more. He then showed some rendering of the Virgin Galactic spacecraft and a logo for the new RED Hydrogen phone which features a 4-view lightfield (like) display. He says the display needs left/right narrow and L/R wide stereo pairs to create the image. His tools can create custom camera arrays that are matched to the display. Figure D.20. Aclectic Light Field Synthesis below shows an example using some of the Aclectic tools.

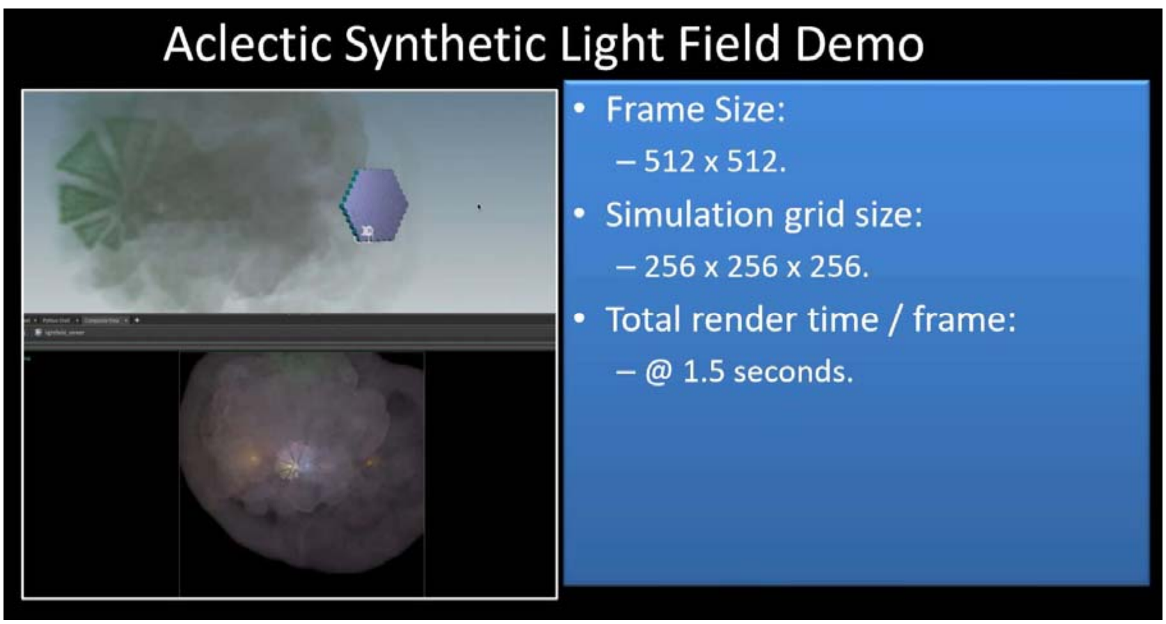

Figure D.20. Aclectic Light Field Synthesis

One particular hardware bottle neck they are addressing is the i/o need for huge datasets. This will initially be based on COTS hardware but will hopefully migrate to FPGAs and software.

D-2.3 Ryan Overbeck, Google, "<u>The Making of Welcome to Light Fields</u>"

Ryan Overbeck from Google delivered a talk that was quite similar to one he gave recently at SIGGRAPH. This described their two light field capture rigs. One is a curved column of 16

GoPro cameras pointed outwards that spins in a circle and the other is a pair of DSLR camera that spin in a spiral pattern. The rigs are good for capturing static scenes and create a navigable volume (user must remain inside this volume) that is less than a meter all around. Nevertheless, they have used the rigs to capture some very compelling scenes like the space shuttle flight deck, Figure D.21 below, and portraits of a couple outside of their colorfully tiled home.

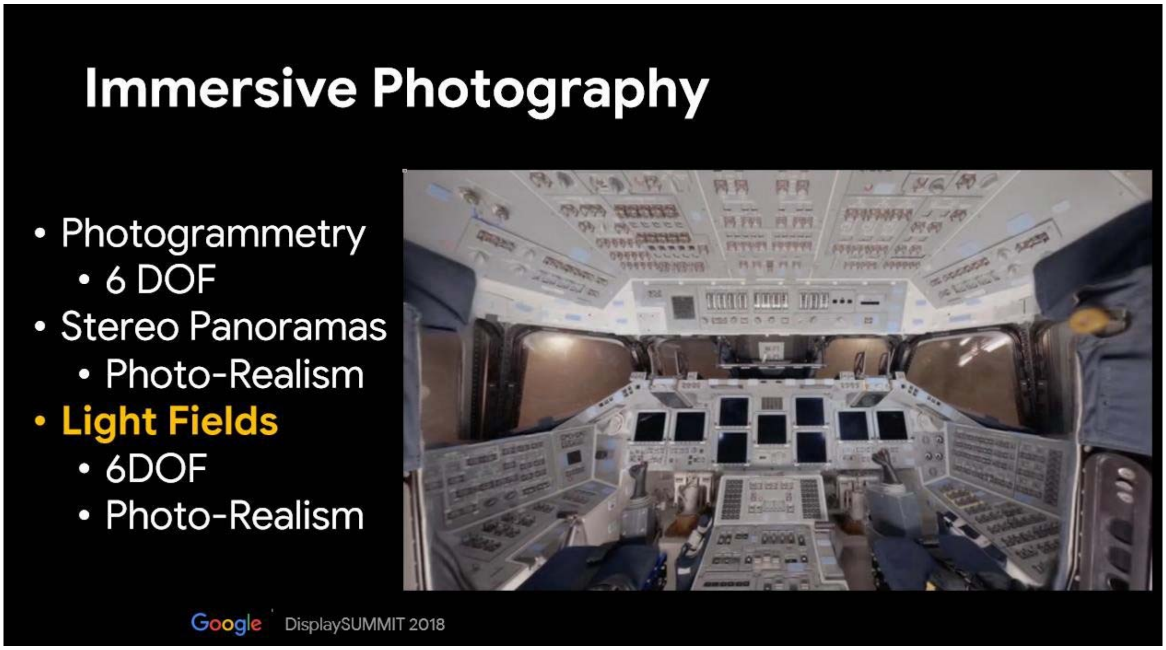

Figure D.21. Interior of Space Shuttle Capture With Google Camera Rig

Overbeck brought one of the rigs to Display Summit and allowed attendees to see this content on a Vive headset. It is clearly some of the best light field content available and it is now available on the Steam VR site along with their SDK. He also described a number of tricks they developed to help improve the images which are amazingly sharp and clear even in the modest resolution Vive headset.

The lightfield data set is rendered in real time to create the stereo images needed for the headset at 90 fps. In development now are methods to extend the concept to capture video.

D-2.4 Arianne Hinds, CableLabs, "Update on Light Field Standards"

Arianne Hinds from CableLabs stated they represent the cable industry and their job is to look at the big picture, understand trends in use cases, and have the infrastructure and technology ready to support new uses as they become viable. She sees light field capture and delivery as one important trend they are preparing for.

She then laid out her assessment of the requirements for commercial adoption of light field systems, as shown in Figure D.22 below. The most important take away is that extending the

current 2D video paradigm to many cameras, i.e. a spatial representation of light field images, is not the way to go. She believes that even live capture with cameras will need to be transformed into a game engine-like model for efficient compression and distribution.
Hinds detailed a number of issues with using a raster approach to light field capture including the lack of a clear "ground truth" to measure various compression schemes against. This was a clear jab at the JPEG PLENO approach which is currently evaluating compression of sparse camera data and comparing it to a ground truth of the original dense camera data.

CableLabs is throwing its support behind a file format called ORBX. This was developed by OTOY as a big "container" to carry all kinds of graphic and special effects data to make it easy to interchange files between facilities. CableLabs plans to work with partner to develop a light field standard around this format and in fact, used Display Summit to announce plans to form a new consortium to further this effort. In passing it is worth noting that the ORBX format is compute heavy and it will likely not be suitable for real time light field streaming.

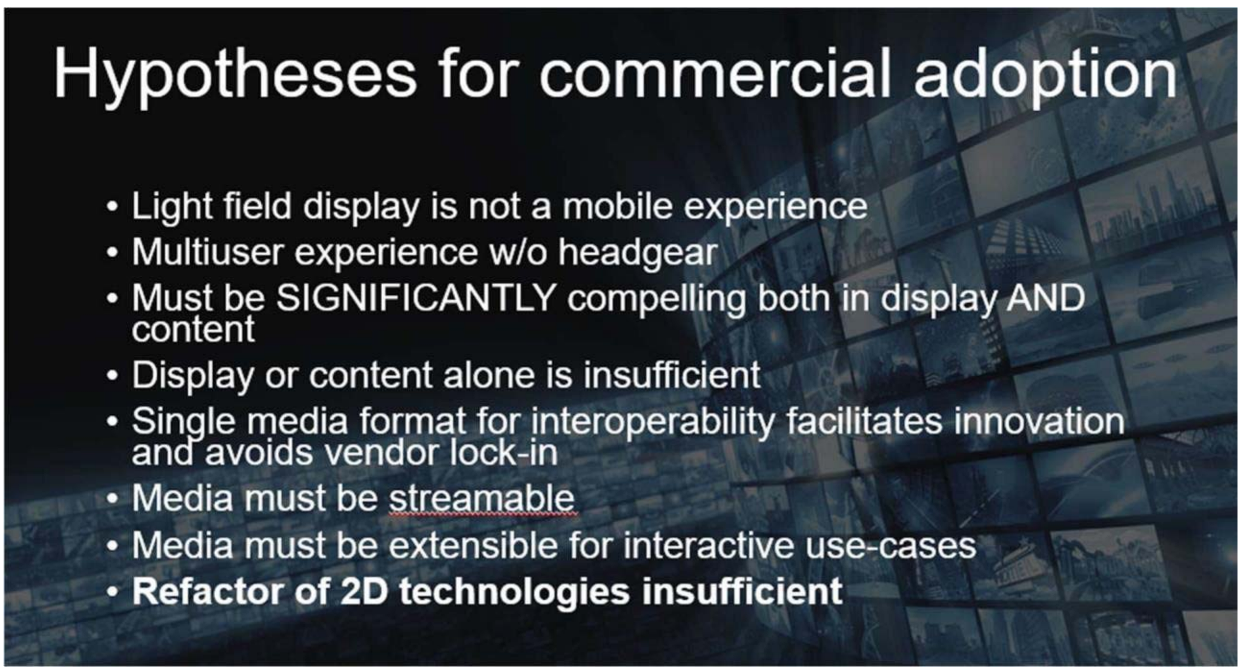

Figure D.22. Cable Labs Hypotheses for Light Field Streaming

D-2.5 Zahir Alpaslan, Ostendo, "JPEG Pleno: A Standard Framework for Representing Plenoptic Modalities"

Zahir Alpaslan from Ostendo then gave an overview of what the JPEG-PLENO group is working on. He started by categorizing advanced images in three groups: ray-based, point clouds and wave-based. Ray-based is what most people refer to as light fields, with point clouds generally used to create depth and/or volume information that might be used in conjunction with other data. Wave-based refers to true holographic images. The JPEG-PLENO effort is aimed at developing a framework that will facilitate the capture, representation and exchange of all three types of images. But, the focus in the initial phase is on ray-based solutions. Figure D.23 below from Ostendo demonstrates schematically the 3D display families.

Most of the work today is focused on developing a codec that can take a series of sparse camera images from multiple cameras or single cameras with microlens arrays, compress/decompress and compare to a high fidelity array of images (at NxN images). This NxN array becomes the Ground truth images that CableLabs thinks is not a real ground truth.

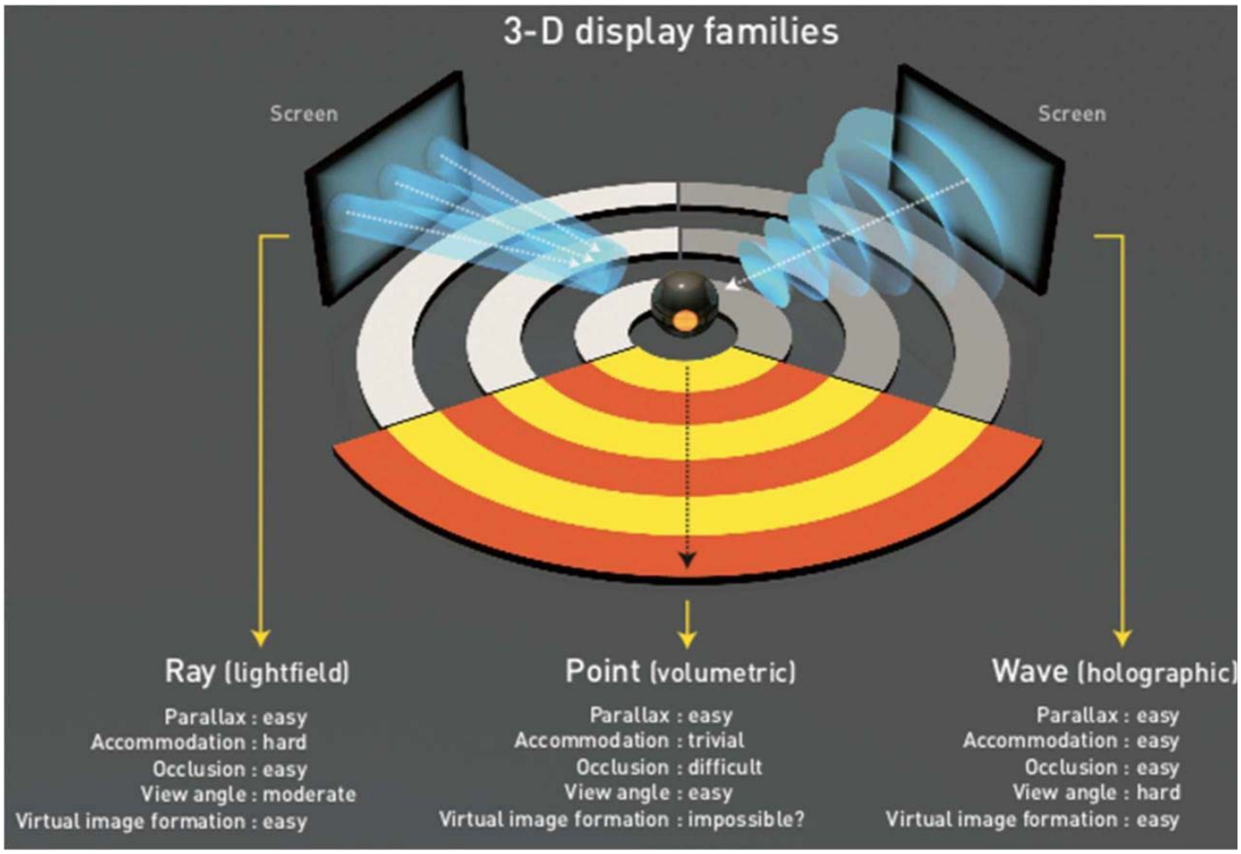

Figure D.23. Various 3D Display Families

Alpaslan noted that the JPEG-PLENO effort supports parallel processing that will likely be needed for tiled light field display solutions (the heterogeneous computing environment later described by FoVI3D). And, they can achieve extremely high compression ratios (he quoted 0.0001 bits per pixel).

The group is currently working to evaluate codecs using subjective (test subjects) and objective (SSIM or Structural SIMilarity) metrics. More test subjects are needed for their experiments. He concluded by noting that point clouds remain immature and that terapixels will be needed to move to true holographic solutions.

D-2.6 Ryan Damm, Visby, "Beyond Lumigraph – Parametric Light Fields for Capture and Display of real-world content"

Ryan Damm is from Visby, a start up in the Light Field space. He started by showing some light field capture solutions but concluded that these cameras "massively under sample," which complicates the representation and delivery part of the chain. His then focused on evaluating several advanced 3D representations to better understand how they perform for compression, delivery and decompression (see criteria in Figure D.24 below). Four representations were evaluated – volumetric, image-based renderer (Lumigraph), BRDF and Parametric Light Field (their area of focus).

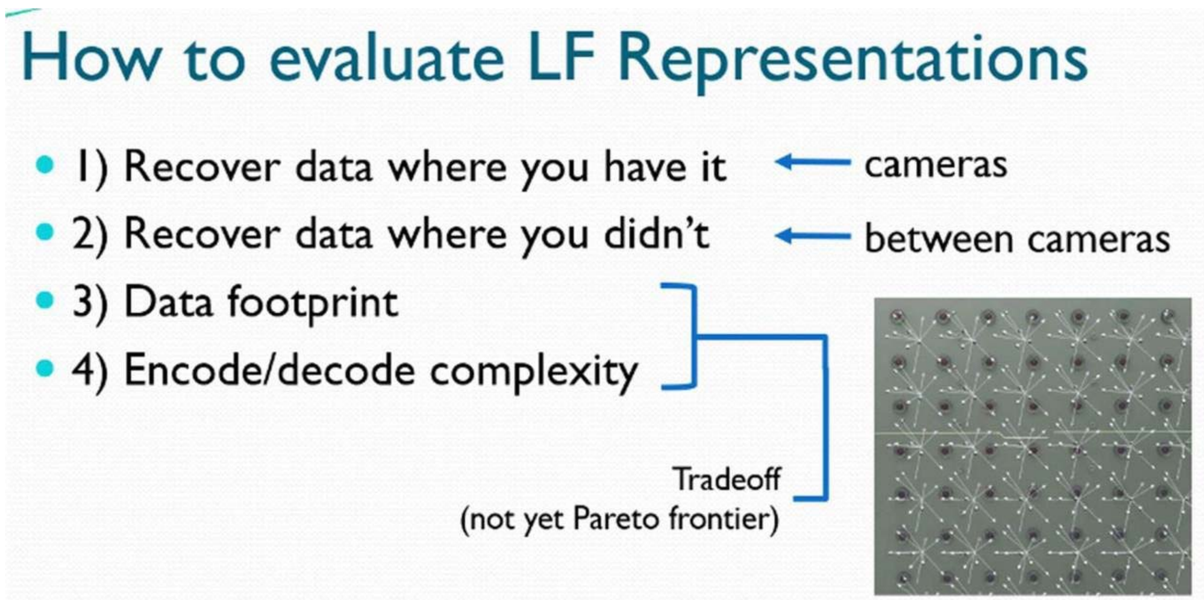

Figure D.24. Criteria for Evaluating 3D Data Representations

Volumetric is good for shape data but pixel data is inherently Lambertian. This approach can only work to create a high fidelity 3D image if you can simulate all the physics of the materials and environment.

Image-based rendering, the so-called lumigraph approach, uses an array of camera of a single camera with a microlens array to capture images. As noted above, it massively under samples the image so reconstruction/interpolation of light rays is needed.

The Bidirectional Reflectance Distribution Function (BRDF) refers to the light reflecting properties of materials. Theoretically, models can be developed of all materials to understand how light from different light sources is reflected in all directions. This can allow for development of complex ray-tracing models to provide a simulation of the light from a scene that includes reflections, transparencies, speculars, etc. It is a massive computing effort and requires complete knowledge of the materials in the scene, but can then be used to quickly relight the scene and/or change materials. Damm admitted he did not know enough about this option to properly evaluate it.

Damm said they are focused on parametric light fields, which he says addresses the deficiencies of lumigraphs (under sampling problems, no representation of smoke, mist or water, lack of true depth data and high data rates). He then tried to explain a parametric light field, but he did not do a good job in explaining it. It seems to mean to use ray tracing using a series of basis functions (with different basis functions for various materials types) to improve the fidelity of the 3D image. Parametric light field will be orders of magnitude smaller in terms of file size and there are no geometries or textures to transmit. Plus, the complexity of the encode/decode is similar to H.264.

D-2.7 Tommy Thomas, Third Dimension Technologies, "Updates on Development of An Open Streaming Media Standard for Field of Light Displays (SMFoLD)"

Tommy Thomas from Third Dimension Technologies (TDT) gave an update on progress they are making on the development of the streaming media standard for field of light displays (SMFoLD). This is an effort funded by the Air Force Research Lab that addresses the need to have a common interchange format for light field data. Clearly, the military has lots of data sets that create 3D images, so having a way to share these more efficiently is a key need.
The TDT approach relies on using OpenGL for the interchange interface, which means light field data is represented as geometries and textures. They propose to develop a customized library that will add some of the functionality needed for light field display light camera position, angle, field of view, etc. This approach allows the source to send one type of data and the display to render it based on its unique capabilities. The display could be 2D, a stereoscopic 3D, horizontal-only parallax multi-view 3D, or a full light field display. Each display would write its own API to allow this to happen. TDT is now in the process of proving that a single source data stream can drive two different 3D display types. Figure D.25 below illustrates the present flow model for SMFoLD.

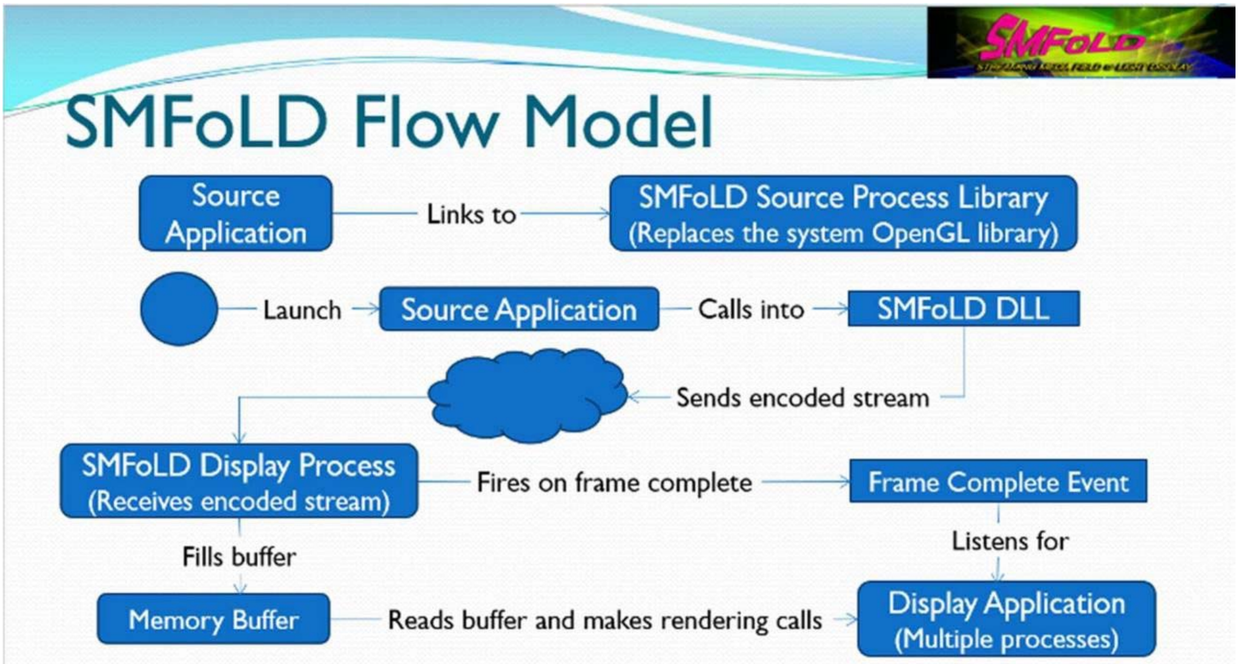

Figure D.25. Flow Model for SMFoLD 3D Frame Creation

D-2.8 Panel Discussion #1

The panel of presenters was asked to discuss what would be needed to move the light field capture and display ecosystem forward. Mr. Overbeck from Google said they would not be scaling up commercial capture until they see a market for light field display devices. Ms. Hinds said that availability of commercial light field displays was the gating factor. Others said that real imagery, as opposed to synthesized imagery, was needed.

The panel discussed standards development efforts. Panel members stated that a standard needed to support both natural and synthetic 3D data. CableLabs is forming a consortium of ORBX users as stakeholders. The ORBX technology would be promoted as an interchange standard, but a streaming protocol would still be needed. The panel was asked about the need for support of a streaming standard by standards bodies. Arianne from CableLabs did not feel that a certifying standards body was necessary. The panel agreed that the Khronos Group should be involved at some level.

D-2.9 Jon Karafin, Light Field Lab, "Even Further Beyond VR: Light Field and Holographic Technology Updates"

In the afternoon session, the focus was more on advanced 3D displays. Jon Karafin from Light Field Labs started the discussion by giving a presentation we have described mostly before. It included a discussion of what he views as a light field display and what is not a light field

display. Many of the 3D displays in movies are not possible, of course, and he says that a true light field display in a head mounted display is not really possible either.
Light Field Labs is developing a tiled-based light field display that can be used for larger-sized display solutions in museums, theme parks and maybe cinemas. The current display module is 6 inches wide and offers 16K x19K pixels, which are used to create light rays in many directions. How it does this is not disclosed. His roadmap calls for commercialization in the 2020 time frame and he even hinted at technologies that may allow interaction for the light field display in the future as well.

D-2.10 Samuel Robinson, Holochip, "Scalable, Real-Time Lightfield Rendering"

Samuel Robinson from Holochip used his time to describe some of the LF display projects they are working on. He divided their efforts into single user and multiple-user projects. Single user displays are less challenging as they can have smaller screens, smaller FOVs and a smaller radiance image resolution. A radiance image can be presented as a 2D image where each pixel represents the color, position and direction of a ray within the light field.
Single user LF displays can use currently available GPU cards (limit 8K per card). The 3D display from JDI, the Leia LF display in the RED Hydrogen phone, LF HMD prototype from nVidia and a flight simulator are all current examples. Robinson said they are working on a helicopter flight simulator for the "chin window" where 3D depth perception is critical for landing, and working with RED and Leia.

Multi-user applications include themed entertainment, cinema and command tables. Here, the FOV, screen size and radiance image all need to be much bigger. This requires new processing architectures and clever ways to deliver data to the LF display. Figure D.26 below lists current efforts under way at Holochip.

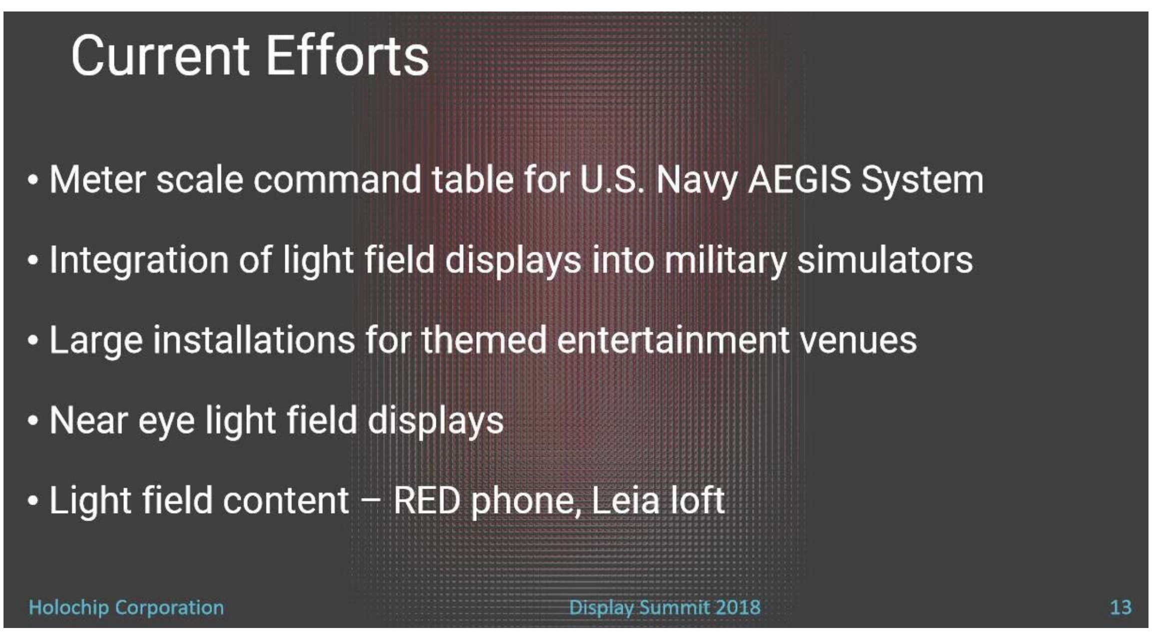

Figure D.26. Current Efforts at Holochip

D-2.11 Thomas Burnett, FoVI3D, "Enabling the Heterogeneous Display Environment "

Thomas Burnett from FoVI3D provided more details on their vision of a heterogeneous display ecosystem. He defines the light field Hogel as "The combination of micro-lens and micro-image. The micro-image colors rays emitting from a point spot on the image plane and the micro-lens angularly distributes the light-rays."

Burnett explained that there are two traditional approaches to computing the radiance image: double frustum and oblique slice and dice. Each has their advantages and disadvantages and they use both methods depending upon the display device and other factors.

Computation of the radiance image needs to be done for each hogel and some of the displays the FoVI3D is working on have 50x50 arrays of hogels. Traditional GPU pipelines render each view in a sequential manner. That means the 50^2 rendering thru the GPU to create just one image. That is not practical, which is why Burnett is calling for a massively parallel approach to computing the radiance image – a heterogeneous computing environment.

Like Third Dimension Technologies, FoVI3D is suggesting a modified OpenGL source code is the way to deliver content that can be played back on multiple types of 2D and 3D displays using only a single source. They are developing an Object Graphic Library for OpenGL (similar to TDT) that would be combined with a parallel computing environment. Their MvPU (Multi-view Processing Unit) concept tries to assign a GPU/CPU to as few hogels as it can to speed up processing.

FoVI3D also showed their latest LF display at the event. Called DevKit Lucas, details are described in Figure D.27 below. This is still a monochrome image with about 4 inches of useable depth before it gets too blurry to be useful. But, the image calibration has improved so that the lines between display modules are now greatly reduced leading to a more uniform image with fewer artifacts.

Figure D.27. Illustration of FoVI3D DevKit 2

D-2.12 Ilmars Osmanis, LightSpace 3D, "Volumetric Displays for Sand Table and HMD Applications: An Update from Light Space 3D"

Switching gears to cover volumetric displays, Ilmars Osmanis from LightSpace Technologies described their technology. It is based on a DLP engine that flashes slices of a 3D data set that are sequentially displayed on a series of diffuser sheets composed of their specially manufactured LCDs. The commercially available model (X1406) is a 19.5 inch monitor with 20 physical depth planes with a 4-inch deep image. It creates an image with 1024 x 768 x 50 voxels of resolution.

A 27" version of this monitor (x2701) is in development and will have 5-6 inches of depth volume offering a resolution of 2560 x 1600 x 40 voxels using 16 physical diffuser planes. Osmanis also described the development of a benchtop display that will have a 39" diagonal and a 5 inch depth volume. Voxel resolution will be 2048 x 1539 x 40 using 8 to 16 physical image planes. They even are developing a 54" version with a 6 inch depth and 5120 x 3200 x 40 voxel resolution. Figure D.28 below illustrates two of the displays under development at LightSpace.

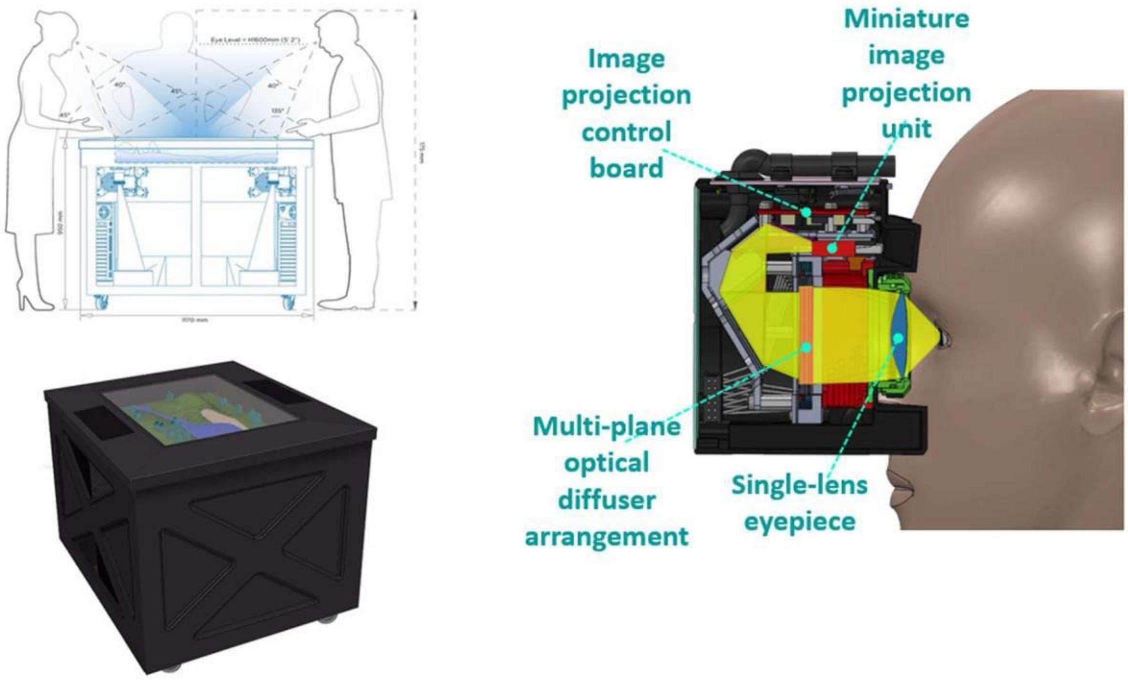

Figure D.28. Displays Under Development at LightSpace 3D

Finally, Osmanis described a project to develop an HMD based on the volumetric concept. He admitted this will be “like a toaster on your head” suggesting it will be big and bulky, but they hope to reduce it if performance is good. LightSpace also showed the 19” model at Display Summit. Images looked quite good due to a new double off-axis DLP engine design that reduces the black level by 100X. It has decent spatial resolution although update rates seemed slow.

D-2.13 Panel Discussion #2

The second panel of presenters was asked to identify the markets where light field displays would likely be adopted, at least initially. The most widely held opinion was that systems would first appear in low volume location-based entertainment applications such as theme parks and arcades. Other applications discussed included DoD for decision-making and high value applications such as air traffic control and medicine. The high cost of the systems and the lack of light field content were cited as hurdles to wide spread adoption.

D-2.14 Discussion of SMFoLD Implementation

TDT discussed the implementation of the SMFoLD standard with a number of display developers and content creators at the workshop. All display developers and content creators that TDT held discussions with indicated that they would implement the SMFoLD standard to gain access to SMFoLD compliant applications.

APPENDIX E - AFRL FOLD WORKSHOPS

E-1 FoLD Workshop – November 15-16, 2017, Wright-Patterson AFB

A two-part FoLD workshop was held at Wright-Patterson AFP on November 15 and 16, 2017 by the AFRL 711th HPW/RHCV, Dr. Darrel Hopper. The first day was dedicated to identifying Government needs and non-proprietary presentations by 3D display industry leaders and researchers. Day two was dedicated to giving researchers and vendors an opportunity to present the state of their research to Government representatives without non-government attendees present. Proprietary information could be revealed on day two. The following is a summary of the first day's presentations (presentation titles only).

E-1.1 Darrel Hopper, Introductions & AFRL FoLD Systems Program, "*True 3D for C2 Ops, Current & Future Topics*"

E-1.2 John Ianni, AFRL, "Space Domain Visualization Challenges"

E-1.3 Nilo Maniquis, Dir, NAVSEA IWS, "*Surface Fleet Combat Information Centers*" Presented by Darrel Hopper

E-1.4 Brian Goldiez, Deputy Director, Inst Sim & Trng, bgoldiez@ist.ucf.edu, *"Human Factors Analysis & Design Guidance for Dynamic Holography in Healthcare"*
Research sponsored by Army RDECOM ARL (Matthew G. Hackett, Mark V. Mazzeo*)*

E-1.5 Michael Cline, NGA, Michael.W.Cline@nga.mil, "T*ools for Effective Use of Layered 3D Geospatial Datasets"*

E-1.6 Arianne T. Hinds, Principal Architect, Cable Television Labs; A.Hinds@cablelabs.com, "*Toward the Development of Distribution Media Standards for Light Field Displays (An Update From Industry*)"

E-1.7 Ilmars Osmanis, "LightSpace3D FoLD Product--Volumetric-Type, Description and Demonstration"

E-1.8 TIPD LLC (TIPD), Massachusetts Institute of Technology (MIT), Brigham Young University (BYU), AF—Phase II *"Holographic Video Display (HVD) Phase II Extension (HVD IIE)"*

E-1.9 Third-Dimension Technologies Inc. (TDT), Oak Ridge National Laboratory (ORNL), Insight Media (IM), AF—Phase II *"Streaming Model for Field of Light Displays (SMFoLD II)"*

E-1.10 Voxtel, Inc., Navy-Phase I "Diffractive Optical Elements for Lightfield Displays (DOE-LFD)"

E-1.11 Holochip, Inc., Navy-Phase I "Light-field Processing Unit for Extreme Multi-View Displays (LFPU-EMVD)"

E-1.12 FoVI3D Inc. Navy-Phase I's (GSD&A, LFPU)

E-1.13 Zebra (dba FoVI3D), AF—Phase II "HVD II and Navy-Full Multiplex Holographic Display Phase II (FMHD II)"

E-1.14 FoVI3D Inc. AF—Phase II "Holographic Lightfield 3D Display Metrology Phase II (HL3DM II)"

E-2 FoLD Workshop – November 14-15, 2018, Wright-Patterson AFB

A two-part FoLD workshop was held at Wright-Patterson AFP on November 14 and 15, 2018 by the AFRL 711th HPW/RHCV, Dr. Darrel Hopper. The first day was dedicated to identifying Government needs and non-proprietary presentations by 3D display industry leaders and researchers. Day two was dedicated to giving researchers and vendors an opportunity to present the state of their research to Government representatives without non-government attendees present. Proprietary information could be revealed on day two. The following is a summary of the first day's presentations (presentation titles only).

E-2.1 Darrel Hopper, AFRL FoLD Systems Program, *"Non-eyewear 3D for BM & C2 Operations"*

E-2.2 Matthew Hackett, USARMY RDECOM ARL , *"Autostereo Displays in Medical and Training Domains"*

E-2.3 LightSpace, *"Time-mux Multi-plane Products, in Production"* (FoLD type: volumetric, mux-2D)

E-2.4 FoVI3D, *"Gen1 & Gen2 Lightfield Display (LfD) Prototypes",* (FoLD type: integral ray, hogel-based)

E-2.5 Third Dimension, *"Holographic Angular Slice 3D Display (HAS3D) Prototypes",* (FoLD type: integral image)

E-2.6 TIPD LLC (TIPD), Massachusetts Institute of Technology (MIT), Brigham Young University (BYU), *"Holographic Video Display Phase II Extension"* (HVD IIE)

E-2.7 Third-Dimension Technologies Inc. (TDT), "*Streaming Model for FoLD*", Year 1 Demonstration

E-2.8 Oak Ridge National Laboratory

E-2.9 Avalon Holographics, Inc., "Holographic Rendering System (HRS) for existing FoLD Systems"

E-2.10 Holochip, Inc., *"Light-field Processing Unit for Extreme Multi-View Displays Phase II"* (LfPU-EMVD II)

E-2.11 FoVI3D Inc., *"FoLD Measurement System, Gen2 Lightfield Display"* (HVD II, FMHD II, HL3DM II)

E-3 FoLD Workshop – November 13-14, 2019, Wright-Patterson AFB

A two-part FoLD workshop was held at Wright-Patterson AFP on November 13 and 14, 2019 by the AFRL 711th HPW/RHCV, Dr. Darrel Hopper. The first day was dedicated to identifying Government needs and non-proprietary presentations by 3D display industry leaders and researchers. Day two was dedicated to giving researchers and vendors an opportunity to present the state of their research to Government representatives without non-government attendees present. Proprietary information could be revealed on day two. The following is a summary of the first day's presentations (presentation titles only).

E-3.1 Darrel Hopper, AFRL Warfighter Interface Division, *"USAF FoLD Systems Program"*

E-3.2 Avalon Holographics, Wally Haas, Mark Newell, *"29-in. Display Prototype Holographic Rendering System"*

E-3.3 Brigham Young University, Dan Smalley, *"AO-EO Waveguide-based Lightfield Display Research"*

E-3.4 FoVI3D, Thomas Burnett, "*Lightfield Display Research & Hogel-based SLM-array LfD Prototypes*"

E-3.5 Holochip, Rob Batchko, *"Lightfield Processing, Collimated Controller Displays, Demo System"*

E-3.6 IntelliSense Systems, Marc SeGall, Tin Aye, *"Hogel-based Picoprojector-array LfD Demo System"*

E-3.7 LightSpace Technologies, Ilmars Osmanis , *"Multi-Depth Plane Volumetric Display Products*"

E-3.8 Third-Dimension Technologies, Tommy Thomas, *"SMFoLD Standard, Angular Slice 3D Prototypes*"

E-3.9 TIPD, Lloyd LaComb, *"Rewritable Holographic Demos & FoLD Processing Research with MIT/BYU*"

E-3.10 Voxtel, Paul Harmon, *"Diffractive Optical Elements for Lightfield Displays"*

APPENDIX F - COMPARISON OF SMFOLD TO OTHER 3D STREAMING MODELS

Several other efforts are underway to create 3D streaming models aimed at the display of light field content. These include MPEG-I, JPEG Pleno, and IDEA.

F-1 MPEG-Immersive (MPEG-I)

The MPEG-I group is working on several different fronts. They are working on 3 Degrees of Freedom (3DoF) encoding which allows the user to look up/down and left/right and head tilt from a single position (pitch, yaw, and roll), 3DoF+ which allows the user to have 3DoF plus limited x, y, z translation, and Six Degrees of Freedom (6DoF) which allows the user to move around freely within a volume.

Coding technologies such as EquiRectangular video Projection (ERP), MultiView + Depth (MVD) Coding, as well as Point Cloud Coding (PCC) for advanced VR/AR and light field display devices are under study. ERP and MVD are common in 3D film production and PCC is a traditional workflow in 3D graphics production. The MVD video coding technologies for MPEG-I are under exploration in the MPEG Video Group, while PCC technologies are studied in MPEG 3DG (3D Graphics Group).[123]

MVD coding work is based on video camera rigs of one form or another, and they take the video and turn it into texture, depth, and metadata. First they prune multiple overlapped images to eliminate the overlapping data, then they also use a form of photogrammetry to obtain depth information from the video, then they turn the nonredundant information into patches for each frame with texture and depth information and an atlas to identify the patch positions. In this manner a view can be synthesized for any allowed position (3DoF, 3DoF+, 6DoF). At this point they are still developing this process. It apparently takes quite a bit of compute time to turn a multi-camera video frame into this format which they call Metadata for Immersive Video. This leaves them with three streams of data to encode—textures (patches), depth and the metadata or atlas of the patch positions.

For MVD coding based on High Efficiency Video Coding (HEVC also known as H.265 compression), it is expected that 0.04 bits per refreshed pixel are needed (including the depth maps), bringing for a typical setup of 16 to 25 camera feeds in UHD (3840 x 2160 pixels), a total of 150-240 Mbps for 30 fps. In applications with Head Mounted Devices requiring much higher frame rates (at least 90-120 fps, i.e. 3 to 4-fold), the total bitrate will increase, but probably less than the corresponding frame rate ratio (expected to be a factor 2).

PCC uses point clouds as data representation and was proposed to find existing codecs that could take advantage of the temporal changes of the data. The point cloud (typically for a single object) is segmented into patches and each patch is projected onto different planes in space with respect to its local orientation, together with its depth maps, and the so-obtained images are coded with traditional 2D video codecs (H.264 or H.265).

The coding performances in PCC rendered on an UltraHigh Definition Display (UHD, 3840 x 2160 pixels) display is about 10-20 Mbps at 30 fps per object on the extensive point cloud animation test set used in MPEG-I Graphics.[124,125]

PCC in MPEG-I Graphics is similar to MVD in MPEG-I Video, there are subtle differences around the use of patches and depth coding.

It does not appear that MPEG-I is presently streaming any 3D data, but they may be ready to in the next year or so. The process appears to be computationally heavy and will likely not allow real-time streaming, but will allow streaming of pre-cached data to displays capable of ingesting it (e.g., VR, AR, or FoLD displays).

F-2 JPEG Pleno

The Joint Photographic Experts Group (JPEG) launched in 2014 the JPEG Pleno initiative to respond to recent developments in the capture, representation, and display of visual information that should be consumed as volumes rather than planes.

JPEG Pleno puts emphasis on three major plenoptic modalities, in particular, light fields, point clouds and holography. "Plenoptic" refers to a mathematical model that considers traditional luminance and color information of any point within a scene, and adds directional information about how this luminance and color change when observed from different positions.[126]

End-to-end plenoptic processing workflow[127] includes data acquisition of light-data from sensors (e.g. 2D camera arrays, depth sensors, devices bearing microlenses) or light-data creation from a computational model. The acquisition/creation process includes metadata that describes the characteristics of the sensors, lenses, or camera arrangements. Metadata can also support semantic search, privacy protection, ownership rights, or security control.

After acquisition/creation stages, a format conversion may be needed to convert the plenoptic data into a suitable representation model due the diversity of sensors, acquisition constraints, or user interaction requirements. An example of a conversion might be transforming a set of texture and depth views into a set of point clouds with RGB values.

Encoding and Packaging stages includes representing the data in a compact way while preserving functionalities such as random access, scalability and error resilience. The encoding process can be either lossless or lossy and applies both to data and metadata. Packaging allows transmission or storage of the light-data.

After transmission or storage, the representation model is recovered through depackaging and decoding, and gets ready for rendering. The rendering technique depends on the representation model, available display and user interaction. Metadata is also considered in the rendering technique in close synchronism with data and, it should be possible to add external models or synthetic content, such as user's hands and body representations when the scene is displayed in immersive displays. The display determines the user experience and technical solutions and requirements for the previous stages of the workflow. User interaction is also relevant to allow fine modifications to the plenoptic content by controlling the point of view, lighting conditions, focal planes, etc.

JPEG Pleno offers a standard framework for the workflow described above. JPEG Pleno is organized in several parts, part 1, 3 and 4, describe details of the framework, conformance tests and reference software respectively, part 2, describes the coding of Light Fields. Three new parts describing the coding of point clouds, holograms and protocols for quality assessment[126] are planned for the near future.

Relevant to this section, part 1 is described in the following. Part 1–Framework, describes how light-field data and metadata is stored and its corresponding file format.[128] The name of this file format is JPL and is based on the JPEG 2000 box-based file format.[129] All information contained in JPL is encapsulated in boxes and stored in binary form. The definition of a specific box type defines the kinds of information that may be found within a box of that type. The box-based structure allows the encapsulation of compressed codestream(s) and any metadata required to describe the encapsulated data, and allows applications to access efficiently the data embedded in the file.

The JPL includes the following boxes.[128] The signature box defines the file is a JPEG Pleno container; the file type box includes information on file type, version and compatibility information. Next, the JPEG Pleno Thumbnail box provides a snapshot of the plenoptic content without need to decode it. The box that encapsulates the light-data is contained in the JPEG Pleno Light Field superbox. Additional boxes provide relevant metadata related to semantic search, privacy protection, ownership rights, or security control.

Of relevance is the JPEG Pleno Light Field superbox. It is organized in a hierarchical way, where the file can contain multiple instantiations of a particular box type.[128] The JPEG Pleno Light Field superbox also contains a JPEG Pleno Light Field Header box that stores parametrization information about the light field (size and color parameters) and coding mode (4D prediction or 4D transform). The Camera parameters box can be optionally used to provide information on the positioning of the local reference grid with respect to the global reference grid, its size and light field’s calibration information. This information is useful when 4D prediction coding mode is used, where intermediate views are predicted from reference frames and its associated depth maps.

Finally, coding mode boxes specify information relevant for 4D prediction and 4D transform modes. The 4D prediction box includes information for the JPEG Pleno Light Field Reference View box that stores compressed reference views of the light field, the JPEG Pleno Light Field Inverse Depth View that includes disparity information for all or a subset of subaperture views, and the JPEG Pleno Light Field Intermediate View box containing prediction parameters and compressed residual signals for subaperture views not encoded as reference views. The 4D transform box is followed by a Contiguous Codestream box that stores the coded light field that uses this mode.

A performance analysis of the 4D prediction and 4D transform coding modes is discussed in Perra, C. et al.[130] More specifically, the performance analysis consisted in comparing metrics such as Peak Signal to Noise Ratio (PSNR), Mean Square Error (MSE) and Structural Similarity index (SSIM) of 4D prediction, 4D transform and HEVC (H.265) codecs. Reported results indicated 4D prediction and 4D transform outperformed HEVC on a set of light field images, representing diversity in terms of acquisition technology, bit depth, spatial resolution, number of views, texture, and scene geometry.

The JPEG Pleno consortium is aimed at encoding static 3D images captured either from multi-camera rigs, plenoptic camera rigs, or holographic (diffractive capture) systems. From encoding static images it may be possible to also encode dynamic video streams. Significant pre-processing is required to create the 3D data so the standard will likely not be suitable for real-time streaming.

F-3 Immersive Digital Experience Alliance (IDEA)

The IDEA consortium (announced in March of 2019) uses a scene graph based container called the Immersive Technologies Media Format (ITMF) for storage and distribution of 3D immersive media created by computer graphics, various volumetric camera arrays, and light field capture methods. ITMF is initially intended as an interchange immersive format of 3D computer generated, audio, and visual media using industry-standard digital content creation (DCC) tools. [9] ITMF will include display specific renderers to allow 3D scene streaming to any display type for which a renderer has been specified and implemented.

Cable Labs and OTOY initially formed as a breakaway from MPEG-I because Cable Labs and OTOY didn't believe in streaming multiple raster video streams, which MPEG-I was working on at the time. The CableLabs and OTOY consortium has now grown into the larger IDEA consortium which entities can join for a fee. IDEA released the first ITMF specification on October 9, 2019. The specification includes three parts: scene graph specification, container specification, and data encoding specification.[131]

The scene graph is a node-based, directed acyclic graph describing logical, temporal and spatial relationships between visual objects in a scene. The scene graph is described in XML format and is based on the ORBX scene graph developed by OTOY Inc. and already in use by a number of visual effects and movie studios.[132] Visual assets that can be referenced in the scene graph are commonly used formats for computer generated imagery and photogrammetry used in design visualization, architecture, visual effects and motion graphics industries.

An ITMF scene graph is composed of one or more nodes, including a mandatory geometry node(s), and a render target node(s). The determination of which nodes must be supported is a work in progress and depends on the target application.

Each node from the scene graph has one or more pins used to create relationships across nodes. A node can have zero or more input pins and have zero or only one output pin. Some nodes may represent 3D objects and have zero or more object attributes that define immutable characteristics of the object. In addition, nodes can have zero or more node attributes that provide metadata to the application specific to the processing for that node.

There are different types of nodes in the scene graph. The render target node determines the final visual output in the serialization processing of the scene graph. The camera node specifies the type of the camera and its attributes present in the scene graph (thin lens, panoramic, baking, etc.). The environment node describes the scene lighting (daylight, texture based light, planetary) and its attributes and source lights are specified in the emission node types. There are different types of geometry nodes depending on how 3D objects are represented (meshes, volumes, points). The physical appearances of 3D objects are specified in the material, medium and transform node types.

The Container part of the ITMF specification defines a packaging system (binary and metadata container) and is based on the container functionality and file structure of ORBX developed by OTOY Inc. The container is comprised of a virtual disk system and a single, high-level XML index file representing the ITMF scene graph. Sections 5 and 6 of the ITMF container specification define the binary markup language and schemas that are used in Section 7 to define the ITMF container's virtual disk system.

The ITMF Data Encoding section of the specification includes the "node code points" to encode the ITMF Scene Graph. After ITMF encoding and containerization, the resulting package becomes transportable across different communication systems and readable by different renderers, digital content creation tools, and display technologies.

IDEA is aimed at the cable and movie industries and appears to be a very heavy-weight standard designed to provide the highest quality image content possible for a given display device. The goal of IDEA is to produce photorealistic 3D imagery with 6DoF for VR, AR, and FoLD displays. Given the tremendous pre-processing required to produce such high quality content in the precise format specified, it is unlikely that the standard will be suitable for real-time use. The method for streaming the 3D content has yet to be identified, however it appears to be reliant on the future availability of 10G networks for streaming to multi-view displays.

LIST OF SYMBOLS, ABBREVIATIONS AND ACRONYMS

1080p	A display with 1920 x 1080 pixels, also known as High Definition
2D	Two-Dimensional
3D	Three-Dimensional
3DoF	Three Degrees of Freedom (angular form MPEG-I, pitch, yaw, and roll)
3DoF+	3DoF plus limited x, y, z translational movement
6DoF	6 Degrees of Freedom, 3Dof plus unlimited x, y, z translation
3GPP	SDO for mobile data
4k resolution	For TV and commercial media, a display with 3840 x 2160 pixels; in the movie industry, a display with 4096 x 2160 pixels
ADX2	Gaming audio middleware from CRI Middleware
AFLMC	Air Force Life Cycle Management Center
AFRL	Air Force Research Laboratory
AIFF	Audio Interchange File Format—a file format used for storing sound data
APE	Monkey's Audi file format—an algorithm and file format for lossless data compression
API	Application Programming Interface
ASIO	Asynchronous Input/Output
ATIS	Satellite data streaming SDO
Alembic	Alembic is an Interchange File Format for Computer Graphics
AOCC	Air Operations Command Centers
API	Applications Programming Interface
ARL	Army Research Laboratories
ARPA-E	Advanced Research Projects Agency-Energy
AWS	Amazon Web Services
BAARS	Battlefield Airmen Augmented Reality Systems
BBOC	Big Ball of Cameras
Blosc	Blosc is a high performance compressor optimized for binary data
BYU	Brigham Young University
CAVE	Cave Automatic Virtual Environment
C2	Command & Control
CAD	Computer Aided Design
CE	JPEG Core Experiment
CfP	Call for Proposals
CIC	Combat Information Center
CID	Center for Information Dominance
Clipmap	Method for clipping a large textured computer dataset
CODEC	Coder-Decoder—generally data compression software
Core Audio	Apple macOS and iOS low-level API for sound programming
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CRT	Microsoft C Run-Time Library
Cryptlib	An open source cross-platform software security (encryption) toolkit library.
DA3DS	Display Agnostic 3D Streaming
DARPA	Defense Advanced Research Projects Agency

DCC	Digital Content Creation
DirectSound	Microsoft Windows sound programming interface
DirectX	Microsoft Windows 3D programming interface
DIBR	Depth Image Based Rendering
DLL, dll	Dynamic Link Library
DoD	Department of Defense
DoE	Department of Energy
DOE	Diffractive Optical Element
DoF	Degrees of Freedom
DoF	Depth of Field
DOPP	AMD Direct Output Post Processing
DTED	Digital Terrain Elevation Data
EDID	Extended Display Identification Data
EMVD	Extreme Multi-View Display
ERP	EquiRectangular video Projection
FCS	Frame Check Sequence
FIPS	Federal Information Processing Standards
FIFO	First In First Out
FLAC	Free Lossless Audio Codec
FMHD	Full Multiplex Holographic Display
FMOD	A cross-platform sound effects engine and authoring tool
FoLD	Field of Light Display
FoLD-PR	FoLD Parallel Rendering
FOV	Field of View
FP	Full Parallax
FPS	Frames Per Second
Gb	Gigabit
GBps	GigaBytes per second
Gbps	Giga bits per second
GitHub	A free repository for sharing Open Source software code
GLUT	OpenGL Utilities Toolkit
GLSL	Open GL graphics shader language
glTF	gl Transmission Format
GLX	OpenGL Extension to the X Window System
GPU	Graphics Processing Unit
GSD&A	TBD
GUI	Graphical User Interface
GXP	Modeling software from the BAE Systems company
HAS3D	Holographic Angular Slice 3D Display
HEVC	High Efficiency Video Coding video compression standard
H.265	Same as HEVC
H.264	A 2D video codec
HL3DM	Holographic Lightfield 3D Display Metrology
HOE	Holographic Optical Element
HP/HPO	Horizontal Parallax/ Horizontal Parallax Only
HRS	Holographic Rendering System proprietary to Avalon Holographics

HVD	Holographic Video Display
IARPA	Intelligence Advanced Research Projects Activity
IBR	Image Based Rendering
IDEA	Immersive Digital Experience Alliance
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IG	Image Generator
IM	Insight Media
Intel IPP	Intel Integrated Performance Primitives
ITMF	Immersive Technologies Media Format
IWS	Integrated Warfare Systems
JON	Job Order Number
JPEG	Joint Photographic Experts Group
JSON	Java Script Object Notation—an open standard human readable file format
LAN	Local Area Network
L1 cache	The high speed first level of cache of a modern CPU
LFD	Light Field Display
LFL	Light Field Lab, Chief Executive Officer is Jon Karafin
LFS	Light Field Streaming
LFPU	Light Field Processing Unit
LiDAR	Light Detection and Ranging
Linux	A Unix based operating system often used on personal computers
LLC	Limited Liability Company
LS	LightSpace
LZ4	An extremely fast lossless compression algorithm
LZ4HC	A higher compression but slower version of the LZ4 algorithm
memcpy()	Memory copy—a computer instruction
MIDI	Musical Instrument Digital Interface
MIT	Massachusetts Institute of Technology
MOD	A computer file format, primarily used for music (MODule file format)
MPEG	Motion Picture Experts Group
MPEG 3DG	MPED 3D Graphics Group
MPEG-I	MPEG-Immersive
MPI	Message Passing Interface
MP3	MPEG Audio Layer 3 Compression
ms	Millisecond
MSS	Maximum Segment Size
MTU	Maximum Transmission Unit
MVC	Multi-view Video Coding
MVD	MultiView + Depth Coding
MVVM	C# Model-View-ViewModel pattern
MVPU	Multi-View Processing Unit
NAVSEA	Naval Sea Systems Command
NDC	Normalized Device Coordinates
NGA	National Geospatial Agency
NIST	National Institute of Standards and Technology

NVAPI	nVIDIA's Core Software Development Kit
ObjGL	Object Graphics Library
OGG	OGG Vorbis compressed audio file, can interleave audio and video
OLCF	ORNL Leadership Computing Facility
OpenAL	Open Audio Language
OpenEXR	Graphics file format
OpenGL	Open Graphics Language—Software API for 3D graphics
OpenVDB	Software Tools for Efficient Management and Storage of Volumetric Data
OPSEC	Operations Security
ORBX	Sophisticated Render Engine and Container for Computer Graphics
ORNL	Oak Ridge National Laboratory
OS	Operating System
OTS	Off the Shelf
PAO	Public Affairs Office
PEO	Program Executive Office
PC	Personal Computer
PCC	Point Cloud Coding
PCI	Peripheral component interconnect
PCIe	PCI express bus
PCM	Pulse code modulation—raw digitized format for audio files
PLENO	Plenoptic Camera
PSNR	Peak Signal to Noise Ratio—method for evaluating graphics encoding
POV	Point(s) of View
QTModeler	LiDAR software viewer
Q&A	Questions and Answers
RGB	Red, Green, Blue
RHCV	AFRL Battlespace Visualization Laboratory
RDECOM	U.S. Army: Research, Development and Engineering Command
RFC	Request for Comments
S&T	Simulation & Training
s2n	An open-source implementation of Transport Layer Security (TLS) protocal
S3D	Stereo 3D
SAR	Synthetic Aperture Radar
SBIR	Small Business Innovation Research
SCTE	SDO for cable streaming
SD&A	Stereoscopic Displays and Applications
SDK	Software Development Kit
SDL	Microsoft Security Development Lifecycle
SDO	Standards Development Organization
SID	Society for Information Display
SIGGRAPH	Special Interest Group on Computer GRAPHics and Interactive Techniques, an annual conference of ACM
SIMD	Single instruction, multiple data
SMFoLD	Streaming Model for Field of Light Displays
SMPTE	Society of Motion Picture and Television Engineers
Snappy	A high speed data compression library

SourceForge	Another free repository for sharing Open Source software code
SOW	Statement of Work
SSE	Streaming SIMD Extensions
SSIM	Structural Similarity—method for evaluating graphics compression
SSL	Secure Sockets Layer—an internet encryption protocol, available as Open Source.
STK	Software Tool Kit for satellite modeling from the AGI company
STTR	Small business Technology Transfer
TCP/IP	Transmission Control Protocol/Internet Protocol, part of Ethernet standard
TBB	Thread Building Blocks
TDT	Third Dimension Technologies
TiGL	Titanium GL, multi-view rendering software interface
TR	Technical Report
TRS	Translate, Rotate, Scale
TLS	Transport Layer Security—another internet encryption standard, open source
Turing Test	Test to Determine if Holographic Image is Indistinguishable from Reality
UDM	Ultrahigh Definition Microdisplay
UHD	UltraHigh Definition Display (3840 x 2160 pixels)
UI	User Interface
UPS	Uninterruptible Power Supply
USAF	United States Air Force
VBA	Vertex Buffer Array
VBO	Vertex Buffer Object
WebSockets	Computer communications protocol proving full duplex communication channels over a single TCP connection
WXGA	A display with a resolution of 1280 x 800 pixels
VAO	OpenGL Vertex Array Object
VBO	OpenGL Vertex Buffer Object
VR	Virtual Reality
Vsync	Forces application frame rate to match up with display frame rate
W3D	Worldwide Web consortium for Internet Standards
WASAPI	Windows Audio Session API
WAV	Waveform Audio File Format—a Microsoft and IBM audio file format standard for storing an audio bitstream on PCs
WiFi	Wireless networking protocol
WireGL	An extension to OpenGL for computer cluster rendering
WGL	OpenGL Extensions for Microsoft Windows
WMA	Windows Media Audio
WPAFB	Wright-Patterson Air Force Base
WSS	Web Socket Secure
WXGA	A video resolution of 1280 horizontal x 800 vertical pixels
Wwise	Wave Works Interactive Sound Engine—Audiokinetic's software features an audio authoring tool and a cross-platform sound engine
XAML	Extensible Application Markup Language file
Zlib	A software library used for data compression
Zstandard	A lossless compression algorithm developed by Yann Collet at Facebook.
Zstd	Same as Zstandard