https://doi.org/10.2352/ISSN.2470-1173.2018.04.SDA-140

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

Jamison R. Daniel, Benjamín Hernández, C.E. Thomas Jr, Steve L. Kelley, Paul G. Jones, and Chris Chinnock

Abstract

Interest in 3D viewing has been increasing significantly over the last few years, with the vast majority of focus being on Virtual Reality (VR), a single-user form of Stereo 3D (S3D) with positional tracking, and Augmented Reality (AR) devices. However, Volumetric 3D displays and Light Field Displays (LFD) are also generating interest in the areas of operational and scientific analysis due to the unique capabilities of this class of hardware. The amount of available 3D data is also growing exponentially including computational simulation results, medical data (e.g. computed tomography (CT), magnetic resonance imaging (MRI), positron emission tomography (PET), ultrasound), computer-aided design (CAD) data, plenoptic camera data, synthetic aperture radar (SAR) data, light detection and ranging (LiDAR) data, 3D data from global positioning system (GPS) satellite scans, and numerous other 3D data sources. Much of this 3D data is available in the cloud and often at long distances from the application or user. While significant progress has been made developing open standards for S3D devices, no standard has yet converged that would allow 3D data streaming for devices such as LFDs that display an assembly of simultaneous views for full parallax and multi-user support without the need for specialized eyewear. A 3D Streaming Standard is desired that will allow display of 3D scenes on any Streaming Media for Field of Light Displays (SMFoLD) compliant device including S3D, VR, AR, Volumetric 3D, and LFD devices. With support from the Air Force Research Laboratories, Third Dimension Technologies (TDT), in collaboration with Oak Ridge National Laboratory (ORNL) and Insight Media, has initialized work on the development of an SMFoLD Open Standard.

Introduction

The hardware and software tools for scientific visualization are evolving rapidly as the computational landscape of high-performance computing continues to expand. Multi and many-core architectures are causing an explosive growth in data sizes and complexities. Similarly, the collection, storage, and transmission of data collected by a variety of sensor systems have increased dramatically over the past 15 years and more rapid growth is anticipated. Traditional approaches of moving source data to remote systems for visualization and analysis are becoming unfeasible and scientists, operators, and decision-makers need to visualize and interpret the data are frequently geographically separated from the source that generates the information.

Server-side transmission of pre-computed visualization results (geometry and renderable objects) to client-side hardware is an approach that reduces the amount of data to transmit while maintaining interactive framerates for remote analysis. This client-side rendering hardware has traditionally consisted of 2D or Stereoscopic (S3D) display technologies. While the advantages of binocular depth perception as a tool for analysis are well established [1], studies have also revealed significant disadvantages to S3D technologies [2]. Eye fatigue and nausea due to a conflict in the accommodation and vergence cues provided to the human visual system is an unpleasant side effect for many users [3]. In order to reduce this discomfort, the motion picture industry leverages techniques involving carefully tailored content delivered to a stationary viewer [4]. However, the unpredictable nature of operational and scientific visualization make these same techniques impractical for real-time workflows. Furthermore, S3D technology 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).

Autostereoscopic displays or so-called Field of Light Displays[1] (FoLD) are glasses-free systems offering full parallax viewing and perspectively correct visualization for multiple persons. These displays leverage the whole human visual bandwidth with little to no visual fatigue. The FoLD class comprises several types including lenticular, volumetric, and holographic technologies [1]. Many existing 3D capture methodologies based on Light Detection and Ranging (LiDAR) sensors, Synthetic Aperture Radar (SAR) sensors, and plenoptic cameras capture a 3D environment that can be viewed correctly from many perspectives only on a FoLD visualization system.

Commercial standards bodies have almost exclusively focused on the 2D and S3D class of displays while the emerging Field of Light hardware technologies has yet to unite behind a common model for streaming a 3D scene description. Unfortunately, the burden of integrating a FoLD system into an application space or environment is repeatedly placed on each software application developer. The resulting proprietary 3D display hardware and software formats limit the adoption and interchange of FoLD visualization devices.

The next step in the evolution of remote 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 flow and POV control as required by the host application. Current and future display prototypes in any class (FoLD, S3D, and 2D) could then create an optimal visualization from the same streaming scene description. Furthermore, development of such an open standard would potentially enhance the speed of development and adoption of various types of high-resolution 3D displays, and particularly Light Field Displays. We are aware that other efforts such as JPEG Pleno and MPEG-I are

[1] From this point, we'll use the FoLD term instead of autostereoscopic

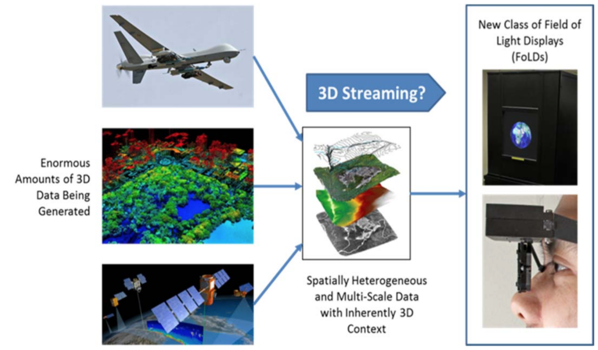

Figure 1. *Example of Command 3D Data Streaming Applications*

in development in this arena, and we will attempt to closely coordinate with these groups as standards are developed. We anticipate that the SMFoLD Open Standard will eventually be turned over to one of the existing National and International standards groups.

Technical Challenges and Opportunities

The scope of the 3D streaming problem is illustrated in Figure 1. Of particular note is the diversity and range of existing solutions and the lack of cohesion. Figure 2 shows a traditional 3D streaming flow. This block diagram begins with the Source, such as a sensor, a computational simulation, or authored content. This Source 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. The Scene and the Network are stages that are not as precisely defined. OpenSceneGraph and CESIUM 3D Tiles offer potential open solutions for the Scene while glTF (gl Transmission Format) is a new open standard from the Khronos Group with interesting potential for the Network. Issues such as latency, bandwidth and view dependence appear under each stage as appropriate.

For the purposes of the SMFoLD Standard, the interchange format will be an SMFoLD OpenGL Frame. The scene will be described by OpenGL primitive and texture information with additional necessary metadata. The metadata will consist of point of view (POV), focal plane, the field of View (FOV) and other data as needed. The network transmission format could potentially exploit glTF. The rendering API for the Display Process will consist of source OpenGL manifested in operational codes (opcodes) which the Display Process can translate to either source OpenGL or an alternative graphics API.

Streaming Challenge

The development of a 3D streaming model has waxed and waned over past few decades beginning with early efforts on VRML in the 1990s, continuing with simulation and training efforts on DIS and CIGI in the 2000s, and more recently since 2009 with the WebGL. While other media formats such as MP3 audio, JPEG images, and H.264 video have converged to recognized delivery standards, 3D standards still remain elusive. The analogy between 3D streaming and other media streaming is tempting but the comparison in Table 1 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. The world of 3D streaming currently consists of a plethora of ad hoc formats and protocols.

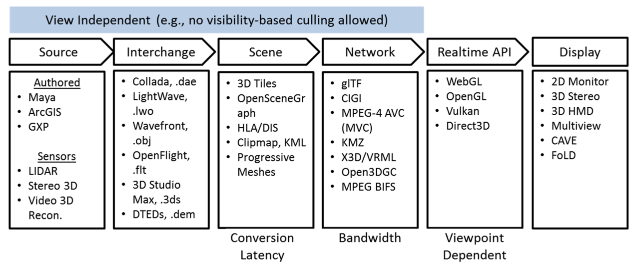

Figure 2. *Block diagram that illustrates the SMoLD flow from source to sink.*

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) and 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 (translate, rotate, scale); also linear key frames possible	Linear key frames
Compression	Difficult to exploit redundancy across mixed heterogeneous data structures	Well-posed redundancy in space and time for signal theory

The comparison in Table 1 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 [5] 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 [6].

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 with no visibility culling allowed and would avoid 2D constructs such as zoom level 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 conceptual paradigm shift in 3D rendering. The SMFoLD standard proposes to resolve this problem by requiring Source Applications to insert metadata into the shader source code and/or the compiled shader code. This metadata will include viewpoint, FOV, data extent, POV and other named variables to be determined. This will allow each rendering pipeline on the Display Application the ability to insert locally named variables for that particular rendering pipeline (viewpoint).

3D Streaming

The diagram in Figure 3 shows a categorization of 3D streaming efforts. Scene streaming involves general techniques that transmit an entire scene for each frame. Some scene techniques are essentially extension of rendering pipelines and include WireGL [7], Chromium [8], and similar methods [9, 10, 11], including work by Oak Ridge National Laboratory [12]. Scene methods also include peer-to-peer concepts [13, 14], and more recently important efforts to create browser-level streaming with WebGL and glTF from the Khronos Group [15, 16]. Google Earth first popularized terrain streaming with the important contribution of Clipmaps [17] with others contributing significant enhancements [18]. The work of Hoppe at Microsoft Research introduced the notion of object streaming with the seminal paper on Progressive Meshes [19]. The last streaming model is image-based rendering (IBR) methods [20, 21], with major efforts on Multi-view Video Coding (MVC) in MPEG standards [5]. Our experience suggests that IBR methods are not a general 3D solution to support the spectrum of FoLD systems [22, 6] . However, the architectural framework of the MPEG committee (with the notable absence of explicit viewpoint in the forward streaming protocols) is important and is similar to Figure 3. 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 [23, 6]. An emerging area that touches each of these categories is cloud-based mobile gaming [24], which has important streaming lessons for user acceptance and adoption [25].

It is our intention that 3D Streaming for the SMFoLD project will consist of SMFoLD frames made up of metadata, OpenGL mesh and texture, and OpenGL primitives.

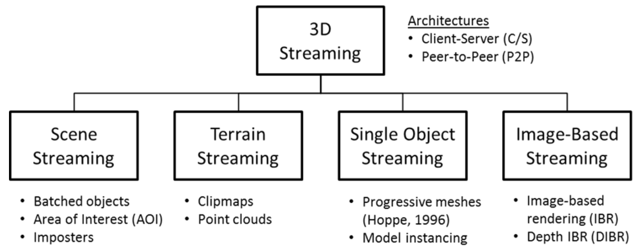

Figure 3. *Categorization of State of Art 3D Streaming Efforts*

3D Display Systems

Over the last 65 years, the work of D. Gabor [26] has inspired serious efforts in developing holographic 3D displays and the associated field of light displays [1, 27, 28]. True diffractive holographic displays (e.g., Massachusetts Institute of Technology (MIT) Media Lab [29, 30, 31] and University of Arizona [38]) are decades away due to pixel size and processing needs [32, 33], but their data requirements should be considered in the development of a new 3D streaming standard.

Aside from the technical challenges, diffractive holograms have more information than is useful to human vision. So researchers long ago developed holographic stereograms, [34, 35, 36, 37] which reproduce all human visual cues [38] with orders of magnitude greater efficiency than diffractive holography. Third Dimension Technology's (TDT) Holographic Angular Slice 3D Display (HAS3D) [39, 40] is an electronic version of a holographic stereogram and thus avoids eye fatigue and display sickness [38, 28] and provides head motion parallax [41] (i.e. look around 3D viewing).

Other 3D display technologies include RealView [42], Holografika [43] and FoVI3D (formerly Zebra Imaging) [44]. 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. More recently Light Field Labs (LFL), has been developing an active pixel system with a wave guide lens to direct the pixels into a 3D light field [45].

XiGen and Physical Optics Corporation [46] also appear to have developed 3D FoLD displays as indicated by several related 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 [47]) displays but these systems have many challenges [38, 27]. Finally, some systems claim to be holographic [47, 48] but are not [49, 1, 28]. Glasses based stereoscopic display systems (including movie theaters and Head Mounted Displays) are also available. These stereoscopic systems introduce convergence accommodation conflicts and make significant portions of the human population sick or uncomfortable.

TDT has been developing FoLD systems since 2003. Figure 4 shows TDTs latest FoLD system (70 diagonal screen and 22 projector illumination) integrated in a flight simulator. TDT developed the simulator under an Air Force Small Business Innovation Research (SBIR) program and delivered the system to the Air Force Research Laboratory (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.

TDTs FoLD technology has a number of advantages for command centers. No glasses are required for 3D viewing while supporting full parallax look around viewing. Moving ones head to see around objects is intuitive and reproduces all human visual cues. Long-term viewing without eye fatigue is another advantage to FoLD technologies. Continuously blended perspectives are present without the need for pseudoscopic active flipping or dead zones, moving mechanical parts, or motion lag associated with traditional tracking systems. This technology can also be used with standard ambient room lighting without the need for light dimming or shades. Compatibility with existing 3D apps (QT Modeler, Google Earth, AGIs STK, GXP) is also an advantage. Scalable design allows for display sizes from desktops to conference rooms to theaters.

Figure 4. *Image of Flight Simulator with Integrated TDT High Resolution FoLD System*

System Overview

Figure 5 shows an overview of the SMFoLD open standard. In general, there is a 3D Source Application and an SMFoLD Source Process running on the source server. The SMFoLD Source Process includes three parts (a) an interface to the 3D Source Application that packages Source Application OpenGL calls, data, and metadata; (b) a complete 3D Frame, and (c) a Codec to compress, optionally encrypt and transmit the 3D Frame. The SMFoLD Display Process includes the codec to decompress/decrypt and reassemble the complete transmitted 3D frame. The SMFoLD Display Process either notifies the 3D Display Application that the 3D frame is ready or waits on a request for the frame. Note that increasing the number of views and/or the resolution of these views increases the quality of the light field while the size of the geometry remains constant.

The combination of a 3D Source Application and the SMFoLD Source Process results in the creation of a 3D key frame similar to a key frame in a 2D streaming application. The SMFoLD workflow begins with defining an interface that a Source Application would use to create the 3D scene. The interface would accept OpenGL function calls and, with the application linked to the SMFoLD library (SMFoLD Source API), the functions it calls will be encoded. Standard OpenGL initialization procedures, such as loading shaders and setting a central view point, would be included. Each function will be assigned an opcode by the standard.

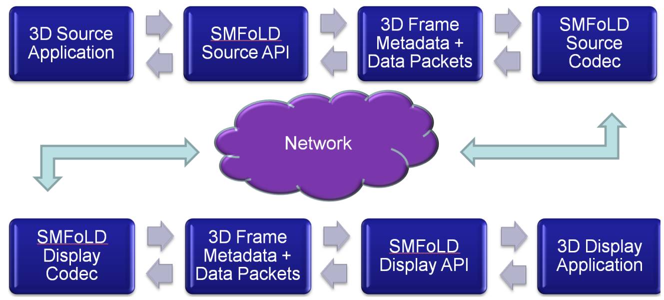

Figure 5. *Streaming Model for Field of Light Displays general diagram*

Preliminary Results

The 3D Source Application must be linked with an SMFoLD library that defines opcodes to replace OpenGL function calls. The library defines an offset into a SMFoLD dynamically linked library (DLL) for each OpenGL function call. The SMFoLD DLL is placed in the Source Application directory. The Source Application does not require additional information associated with the opcodes and is concerned only with the execution of the original OpenGL calls.

Figure 6. *SMFoLD Source Process Implementation*

Additional functions in the SMFoLD Source Process (smfold.dll) will create additional metadata needed to complete an SMFoLD compliant scene. The OpenGL calls in the Source Application that indicate the rendering buffer is ready to display will be used to signal the end of a 3D frame. This approach supports porting existing applications to SMFoLD compliance with minimum effort.

The modern development of graphics application commonly involves the creation of shaders. Shaders provide more dexterous instructions for the rendering pipeline than traditional OpenGL state function calls. These unique instructions can execute during several different stages in the graphics pipeline. Of particular concern to our proposed model is the vertex shader. The vertex shader is responsible for converting the input coordinates into normalized device coordinates. In order to support various displays, it will be necessary for vertex shaders to include a call to an SMFoLD defined function that will adjust the output of the vertex shader so that it is correct for the viewpoint being rendered. A shader header file will be created as part of the standard definition that will contain a shader function that performs the geometric adjustment. The header file will also include variable definitions that will allow the display application to control the parameters used for the adjustment. The variables will be defined to allow the display application to set an offset and rotation for a unique camera perspective for each view the display generates. These calculations are performed in normalized device coordinates space, thus no display specific information is necessary. SMFoLD compliance will require all 3D Source Application vertex shaders to include this SMFoLD defined header. This header includes the additional geometry variables such as camera position, camera angle, FOV, and others as deemed necessary and also defines the associated function calls that generate the additional information needed by the FoLD display hardware. The geometry variables are available to the Display Application and must be initialized for each viewpoint that uses the shader.

Shown below in Figure 16 is the format of a typical SMFoLD 3D data frame.

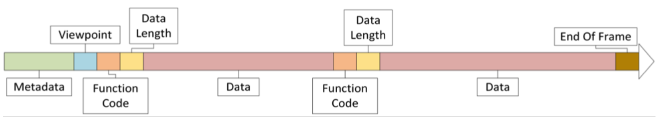

Figure 7. *Format of a Typical SMFoLD 3D Data Frame*

Backwards compatibility with unmodified legacy OpenGL applications can be accomplished if the original vertex shader source code is available. Replacing the dynamically linked OpenGL library with an identically named SMFoLD library allows the original vertex shader code to be parsed and modified to include the required named variables and geometry function calls. However, full SMFoLD functionality will be limited. For example, communication from the Display Application back to the Source Application will not be possible.

To further reduce bandwidth requirements, a mechanism is needed to allow a Source Application to change individual values within a reusable resource without having to download the resource in its entirety. Only data within a frame that is changed needs to be streamed. A complication exists from the possibility that the data in GPU memory buffers is modified directly by the CPU. These changes are made by the CPU outside of the graphics API and the Display Process is not aware that the changed has occurred. The solution currently is to re-transmit all of the data. However, a more efficient method would require the Source Application to use SMFoLD API extensions to reference and delineate the changes. The Source Application would then send only the changed data along with the necessary addressing information, thus allowing the Display Application to apply the changes. This change frame technique is similar to the change frame implementations in 2D streaming.

The SMFoLD Source Process creates metadata that allows the Display Application to create in-focus views. Examples of metadata include geometry limits (extents of the data), viewpoint changes on the FoLD display to create multiple viewpoints, information needed to create the desired amount of parallax, and areas of interest. This metadata can be defined either by the Source or Display Application. Additional metadata will likely be added as the SMFoLD standard is implemented.

Once the frame is complete and the SMFoLD Source Process has been notified, the memory buffer is ready to be processed for transmission. See Figure 8. Processing will include compression and optionally encryption by the SMFoLD Source Codec. Open source compression and encryption algorithms are available and currently being considered for inclusion into the SMFoLD standard.

Figure 8. *SMFoLD Frame Transmission*

For compression we are investigating algorithms including Lossless, Run-Length Encoding, LZ, MPEG 3DGC (connectivity), 3D Point Cloud Compression, Lossy, floating point conversion, MPEG 3DGC (geometry), and JPEG (textures). The computation overhead of mesh compression is not trivial, however, and will likely be handled separately. Lossy mesh compression implemented by converting 32-bit floating point mesh location coordinates to 16-bit integers offers an immediate reduction of the mesh data without significant loss of accuracy since there are not yet any displays with 65,000 pixels in one row or column. Table 2 demonstrates potential frame rates and corresponding compression rates for 3D Frames from various sample applications with compression ranging from 25 percent to 75 percent.

Frame Rates (FPS) with 1 Gbps (125MBps) Speed Internet at Different Compression Level C

Application	Data Size (Bytes)	C (25%)	FPS	C (50%)	FPS	C (75%)	FPS
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

Encryption would be optionally performed on request by the Source Application codec. Several encryption libraries are available and most have licenses that would facilitate implementation into the SMFoLD standard. The AES method is currently the method of choice for most new applications and is the standard trusted by the U. S. Government. [50]

Once a frame has been pre-processed it is ready for transport. The SMFoLD Source Process will pass the data from the memory buffer to the network interface using standard TCP/IP protocols.

As shown in Table 2 above, frame rates on the example applications are reasonable with high network speeds. In situations where the network speed is slower, frame rates may cause the FoLD system images to update at an undesirable and visibly detectable rate. Significant degradation to the audio signal may also occur. In this case, higher compression ratios must be achieved potentially at the loss of resolution, network bandwidth must be increased, or quality degradation must be tolerated. This must be addressed in the Source Application if additional network bandwidth is not available. Since the 3D Frame uses mesh and texture data and other OpenGL graphics primitives, the Source Application can potentially reduce the complexity of the 3D Frame until an acceptable frame rate is achieved. Backwards communication (either automatic or by an operator) from the SMFoLD Display Process will be required to implement such a solution. Note that bidirectional communication between the SMFoLD Display Process and the SMFoLD Source Process is already a feature of the proposed standard.

An SMFoLD Display Process (See Figure 9) on the receiving end of the data stream listens on the designated network interface port for incoming data. When the transmission is complete the SMFoLD Display Process codec will decrypt the data if it is encrypted and decompress it. The results of the decompression will be written to a shared memory buffer defined by the SMFoLD standard. When this process is complete it will signal the Display Application that a frame is ready to be processed.

Figure 9. SMFoLD Display Process Implementation

Summary

We have developed a conceptual design for a display agnostic standard for streaming 3D graphics known as Streaming Model for Field of Light Displays (SMFoLD). The proposed standard will allow SMFoLD compliant displays to receive a stream of 3D frame descriptions and render a 3D scene using a consistent and repeatable methodology. The computer driving the display device must be running an SMFoLD compliant display application. Recall Figure 1 that represents a typical command center workflow 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.

Two SMFoLD workshops have been held by Third Dimension Technologies (TDT). The first workshop was held in October 2016 with significant help and assistance from Insight Media and the Society of Motion Picture and Television Engineers (SMPTE). This workshop covered the entire ecosystem required for FoLD displays. One of the main conclusions of the workshop was that there should be an SMFoLD standard and work should continue to develop this standard. Another SMFoLD workshop was held in October of 2017 at a Rockwell Collins site in Sterling, VA, in conjunction with the Display Summit 2017 conference. Again there was significant industry and government attendance. Both workshops show that there is strong interest in the development of a display agnostic 3D streaming model in both government and industry. To further explore the material presented at these workshops and the summary discussions, the reader is invited to visit the SMFoLD website, which has presentations and summaries from both workshops [51].

Data show that a natural 3D view improves peoples performance when making judgments about the spatial relationships among objects in a scene. The enhanced performance can help operators in Command and Control centers understand situations in the battle space, provide a naturally intuitive and collaborative environment to analyze scientific data, improve learning for medical trainees, enable architects to create structures that utilize space more efficiently, and create environments that reduce the need for travel. The development of a successful SMFoLD standard can hasten the availability of interchangeably compatible FoLD systems. In addition, SMFoLD compliant Source Applications can be run locally on SMFoLD compliant displays without the need for streaming 3D data, thus the standard provides interoperability of future 3D software applications with SMFoLD compliant displays.

Creating a usable standard depends on a methodology that overcomes the tremendous data sizes needed for Plenoptic approaches and provides display developers all of the information necessary for their displays to create 3D scenes. The standard proposed here starts with a simple implementation using the existing OpenGL API standards that will create relatively small 3D Frames suitable for network transmission at potentially acceptable 3D Frame display rates.

This proposed SMFoLD standard is partially based on the successful deployment of TDT's Horizontal Parallax Only (HPO) OpenGL based system, Holographic Angular Slice 3D Display (HAS3D). Our effort builds on that success by creating a standard that will be supported relatively easily. The idea is to start simple using a known paradigm and leave the door open for future improvements and additions. By creating a working demo using the HAS3D system we will be able to demonstrate the standard to Government and Industry leaders thus accelerating adoption and wide spread use.

Acknowledgments

This work was funded by AFRL contract FA8650-17-C-6877, which was selected under the STTR program entitled " AF16-AT07 Streaming Model for Field of Light Displays (SMFoLD). This paper is DISTRIBUTION STATEMENT A. Approved for public release: distribution is unlimited. 88ABW Cleared 12/21/17; 88ABW-2017-6434.

References

[1] N. S. Holliman, N. A. Dodgson, G. E. Favalora and L. Pockett, "Three-Dimensional Displays: A Review and Applications Analysis". IEEE Transactions on Broadcasting, vol. 57, no. 2, pp. 362-371, June 2011.

[2] Urvoy, Matthieu. Barkowsky, Marcus and Le Callet, Patrick, "How visual fatigue and discomfort impact 3D-TV quality of experience: a comprehensive review of technological, psychophysical, and psychological factors". Annals of Telecommunications - Annales des Télécommunications, 68, 11 (2013)

[3] David M. Hoffman, Ahna R. Girshick, Kurt Akeley, Martin S. Banks, "Vergence–accommodation conflicts hinder visual performance and cause visual fatigue". Journal of Vision 2008;8(3):33

[4] Solimini AG, "Are There Side Effects to Watching 3D Movies? A Prospective Crossover Observational Study on Visually Induced Motion Sickness". PLoS ONE 8(2): e56160.

[5] S. Ince, E. Martinian, S. Yea and A. Vetro,"Depth Estimation for View Synthesis in Multiview Video Coding". 2007 3DTV Conference, Kos Island, 2007, pp. 1-4.

[6] Stephan Würmlin , Edouard Lamboray , Markus Gross, "3D video fragments: Dynamic point samples for real-time free-viewpoint video" Comput Graph 28.1: 3-14. 2004.

[7] G. Humphreys, I. Buck, M. Eldridge and P. Hanrahan, "Distributed Rendering for Scalable Displays". Supercomputing, ACM/IEEE 2000 Conference, 2000, pp. 30-30.

[8] Greg Humphreys, Mike Houston, Ren Ng, Randall Frank, Sean Ahern, Peter D. Kirchner, and James T. Klosowski. "Chromium: a stream-processing framework for interactive rendering on clusters".

Proceedings of the 29th annual conference on Computer graphics and interactive techniques (SIGGRAPH '02). ACM, New York, NY, USA, 693-702.

[9] Eilemann S, R Pajarola, "The Equalizer parallel rendering framework". Technical Report. Univ Zurich 2007.

[10] T Ilmonen, M Reunanen, P Kontio,"Broadcast GL: An alternative method for distributing OpenGL API calls to multiple rendering slaves" Journal of WSCG, 13(2), pp. 65-72, 2005.

[11] Lorenz, M, G Brunnett. "Optimized visualization for tiled displays". Eurographics Symposium on Parallel Graphics and Visualization (EGPGV), 2006.

[12] Paul, B, et al. "Chromium Renderserver: Scalable and Open Remote Rendering Infrastructure". IEEE Transactions on Visualization and Computer Graphics, vol. 14, no. 3, pp. 627-639, May-June 2008.

[13] Hu, S-Y. "A case for 3d streaming on peer-to-peer networks". ACM International Conference of 3D Web Technology. 2006.

[14] Vani, V. "New approaches for 3D streaming and rendering based on predictive modeling." 2014.

[15] A. Evans, M. Romeo, A. Bahrehmand, J. Agenjo, J. Blat. "3D graphics on the web: A survey". Computers Graphics 41: 43-61. 2014.

[16] glTF. Runtime asset format for WebGL, OpenGL ES, and OpenGL. https://www.khronos.org/gltf. 2014.

[17] Christopher C. Tanner, Christopher J. Migdal, and Michael T. Jones. "The clipmap: a virtual mipmap". Proceedings of the 25th annual conference on Computer graphics and interactive techniques (SIGGRAPH '98). ACM, New York, NY, USA, 151-158.

[18] Losasso, F, H Hoppe. "Geometry clipmaps: terrain rendering using nested regular grids." ACM TOG 23, 2004.

[19] Hoppe, H. "Progressive Meshes." SIGGRAPH. pp. 99-108, 1996.

[20] 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.

[21] Smolic, A, P Kauff. "Interactive 3D video representation and coding technologies". Proc IEEE 93:98-110. 2005.

[22] Lamboray, E, et al. Real-time streaming of point-based 3D video. IEEE VR, 2004.

[23] Jurgelionis, A, et al. Platform for distributed 3D gaming. Intl J Comput Games Tech. 2009.

[24] Dinh, HT, et al. A survey of mobile cloud computing: architecture, applications, and approaches. Wireless communications and mobile computing 13.18: 1587-1611. 2013

[25] Kruger, M. The future of cloud gaming? Wired. Aug 2013.

[26] Gabor, D. Microscopy by Reconstructed Wavefronts, Proc Roy Soc. London Ser A, A197, p. 459, 1949.

[27] Konrad, J, M Halle, 3-D displays and signal processing: An answer to 3-D ills? IEEE Sig Process Mag 2007.

[28] Page, DL, et al., Vergence and accommodation in simulation and training with 3D displays. I/ITSEC. 2014.

[29] Bove, VM. What is Holographic Television, and Will it Ever be in My Living Room? SMPTE Intl Conference on Stereoscopic 3D for Media and Entertainment, 2010.

[30] Smalley, D, et al. Holographic video display based on guided-wave acousto-optic devices, SPIE, 6488. 2007.

[31] Smithwick, Q, et al. Interactive holographic stereograms with accommodation cues, SPIE Practical Holography XXIV, Vol. 7619, 2010.

[32] Halle, M., Autostereoscopic displays and computer graphics, SIGGRAPH, 1997.

[33] Slinger, C, C Cameron, M Stanley, Computer-generated holography as a generic display technology, Computer, 38(8), pp. 46-53, Aug 2005.

[34] DeBitetto, D., Holographic panoramic stereograms synthesized from white light recordings, Applied Optics, 8(8), pp. 1740-1741, 1969.

[35] Halle, M, Holographic stereograms as discrete imaging systems, SPIE, Vol. 2176, 1994.

[36] King, M, et al. A new approach to computer-generated holography, Applied Optics, 9(2), pp 471-475, 1970.

[37] Klug, M, et al. Optics for Full-Parallax Holographic Stereograms, SPIE Pract Holo XI, 3011, pp 78-88, 1997.

[38] Blanche, P-A, Toward the ultimate 3-D display, Information Display, 28(3), 2012.

[39] Thomas, CE, System Methods for Angular Slice True 3-D Display, US Pat. 7,513,623 B2, Apr. 7, 2009.

[40] Thomas, CE, System Methods for Angular Slice True 3-D Display, US Pat. 8,075,138 B2, Dec. 13, 2011.

[41] Cutting, J, How the eye measures reality and virtual reality, Behavior Research Methods, Instruments Computers, 29(1), pp. 27-36, 1997.

[42] RealView Imaging Ltd., www.realviewimaging.com, Accessed on Feb. 5, 2016.

[43] Balogh, T, et al. An interactive multi-user holographic environment, SIGGRAPH, 2006.

[44] Zebra Imaging, www.zebraimaging.com, Accessed on Feb. 2, 2016.

[45] Light Field Lab, http://www.hollywoodreporter.com/behind-screen/nab-consumer-ready-holograms-move-closer-reality-997242 , accessed on 8 Sept 2017

[46] Physical Optics Corporation, http://www.poc.com, Accessed on Jan 29, 2016.

[47] Setred, www.setred.com, Accessed on Jan 30, 2016.

[48] zSpace, www.zspace.com, Accessed on Jan. 30, 2016.

[49] Brennesholtz, M. Provision looking for crowd-source funding on Kickstarter, Display Central, July 2013.

[50] Hinds A., "Toward MPEG's Vision of Immersive Experiences", SMFoLD Conference, November 2016, http://www.smfold.org/wp-content/uploads/2016/11/MPEG-for-SMFoLD.pdf

[51] Streaming Model for Field-of-Light Displays (SMFoLD). http://www.smfold.org/. Accessed on December 18, 2017

Author Biography

Jamison Daniel is a research scientist in the Advanced Data and Workflows group at the Oak Ridge Leadership Computing Facility (OLCF) at the Oak Ridge National Laboratory (ORNL) where he leads the deployment of advanced deep learning technologies, the development of remote visualization workflows, and provides data liaison support for the high-performance computing projects running on the Titan supercomputer.

Benjamín Hernández is a computer scientist at Oak Ridge National Laboratory serving on the Advanced Data and Workflows group. He collaborates with TITAN supercomputer users to enable scientific insight through the development of scalable parallel data and visualization solutions. He holds a Ph.D. in Computer Science from Tecnologico de Monterrey, Mexico. His research interests are in the intersection of interactive computer graphics, human computer interaction and high performance computing.

C. E. (Tommy) Thomas is the founder and Chief Technical Officer (CTO) of Third Dimension Technologies (TDT) and has been working with holography, lasers and electro-optics design for more than 30 years (since 1972). Dr. Thomas holds a PhD in Plasma Physics (holography the-

sis) from Massachusetts Institute of Technology (MIT), Cambridge, MA, 1980, and is the inventor of the Holographic Angular Slice 3D Display (HAS3D).

Steve L. Kelley has over 25 years of programming experience developing user applications and creating software for embedded systems. He was the lead programmer of a scientific graphics library for Scientific Endeavors where he developed algorithms for high speed rasterization techniques for graphics cards and printer drivers. Mr. Kelley is the Director of Software Engineering at Third Dimension Technologies and has led the efforts to optimize the 3D rendering pipeline for holographic display technologies.

Paul G. Jones is Director of Marketing for Third Dimension Technologies (TDT). Prior to TDT, Mr. Jones was the Chief Marketing Officer for nLine Corporation, a semiconductor equipment company he cofounded in 1998. Mr. Jones spent many years in the semiconductor industry beginning his career at Texas Instruments. He worked as a process engineer and manager at Advanced Micro Devices and industry consortium Sematech before becoming marketing director for inspection products at Electroglas.

Chris Chinnock is the President of Insight Media (www.insightmedia.info), located in Norwalk, Connecticut. Chris holds a BSEE from the University of Colorado and founded Insight Media in 1998. Prior to this, he worked as a freelance journalist and consultant plus held a variety of technical and management positions with General Dynamics, Honeywell Electro Optics, MIT Lincoln Labs, General Electric Astro Space and Barnes Engineering.