Navigable Videos for Presenting Scientific Data on Affordable Head-Mounted Displays

Jacqueline Chu, Chris Bryan, Min Shih, Leonardo Ferrer, Kwan-Liu Ma
Visualization & Interface Design Innovation (VIDi) Labs
University of California, Davis
{sjchu, cjbryan, minshih, ferrer, klma}@ucdavis.edu

ABSTRACT

Immersive, stereoscopic visualization enables scientists to better analyze structural and physical phenomena compared to traditional display mediums. Unfortunately, current head-mounted displays (HMDs) with the high rendering quality necessary for these complex datasets are prohibitively expensive, especially in educational settings where their high cost makes it impractical to buy several devices. To address this problem, we develop two tools: (1) An authoring tool allows domain scientists to generate a set of connected, 360° video paths for traversing between dimensional keyframes in the dataset. (2) A corresponding navigational interface is a video selection and playback tool that can be paired with a low-cost HMD to enable an interactive, non-linear, storytelling experience. We demonstrate the authoring tool's utility by conducting several case studies and assess the navigational interface with a usability study. Results show the potential of our approach in effectively expanding the accessibility of high-quality, immersive visualization to a wider audience using affordable HMDs.

CCS CONCEPTS

•**Human-centered computing** → **Virtual reality;** *Scientific visualization;* •**Applied computing** → *Interactive learning environments;*

KEYWORDS

immersive visualization; stereoscopic visualization; head-mounted displays; scientific storytelling

ACM Reference format:
Jacqueline Chu, Chris Bryan, Min Shih, Leonardo Ferrer, Kwan-Liu Ma. 2017. Navigable Videos for Presenting Scientific Data on Affordable Head-Mounted Displays. In *Proceedings of MMSys'17, Taipei, Taiwan, June 20-23, 2017,* 11 pages.
DOI: http://dx.doi.org/10.1145/3083187.3084015

1 INTRODUCTION

Computational and applied science domains use visualization to explore, analyze, and present the complex 3D phenomena and structures that make up our universe. Using data from large-scale, multidimensional simulations, conventional plotting is traditionally done using 2D display monitors with graphics techniques like perspective, lighting, shading, clipping planes, and transfer functions. Compared to purely flat displays, *stereoscopic visualization* improves perception of the shapes, geometries, volumes, and relationships that exist in the data [25]. Virtual reality (VR) and other immersive-based techniques such as HMDs (head-mounted displays) with head-tracking support are the current state of the art in wide-view, stereoscopic visualization. Unfortunately, high-quality display and interaction in this context is non-trivial due to many factors: the data is large and complex, requires high-precision, and often must be stored using more expensive volumetric structures (as opposed to 3D meshes).

This results in a quandary– in cases where interactive, immersive, stereoscopic visualization is the optimum choice for a scientific dataset, it should be the chosen presentation medium. But effectively visualizing the data (i.e., with high quality, high frame rate, and low latency) is likely predicated on using expensive rendering algorithms. If money and space is not an issue then high-end HMDs or CAVEs may be used. However, in many cases this is not a realistic solution due to a combination of required space, monetary costs, and/or hardware availability. This is especially true in educational contexts like classrooms or museums. Alternatively, using more affordable, lower-end HMDs with standard rendering techniques can lead to poor performance [46]. Consequently, current solutions require making compromises such as data compression and sub-sampling, low rendering quality [11], or limited interaction [33].

Sacrificing visualization quality in presentation and teaching environments is a poor solution because it detrimentally affects how the content is perceived and thus learned. Similarly, forcing high rendering quality while accepting low frame rates or high latency (defined as the time between when a user moves his/her head to when the image is updated, also known as motion-to-photon [21]) can make for a frustrating user experience.

Our solution to this problem is the development of two tools. The *authoring tool* allows for creating and subsequently exporting interconnected, pre-rendered, 360° navigable videos of a scientific dataset. The *navigational interface* lets a user traverse and view the set of navigable videos. This playback tool is usable on low-end HMDs, thus providing an interactive, immersive, high-quality, storytelling experience of a complex, visually expensive dataset.

Specifically, in the authoring stage a domain scientist can load his/her dataset into the authoring tool. The data is rendered at high quality using a suitably powerful desktop machine and graphics card. The scientist creates a *roadmap* of connected videos. These are keyframed animation sequences that interpolate dimensional changes in the dataset, such as changing the camera's position or manipulating the transfer function. Animation sequences act as edges in a graph; shared and terminal keyframes act as nodes. Exporting

MMSys'17, Taipei, Taiwan
© 2017 ACM. 978-1-4503-5002-0/17/06…$15.00
DOI: http://dx.doi.org/10.1145/3083187.3084015

the roadmap renders a set of high-quality, stereoscopic videos– one video for each sequence. Loaded into the navigational interface, the set of videos can be selected and viewed using an HMD (or other immersive hardware setup).

This approach is similar in concept to existing, 2D, picture-based navigation interfaces (such as Google's Street View) which allow a user to jump between different 360° photographs. In our case, instead of showing individual panoramic images that are jumped between, the authoring tool creates immersive video "tubes" that can be seamlessly and bi-directionally played and navigated.

As opposed to having the HMD naively render the full dataset, our approach has multiple advantages for data presentation and teaching contexts. Showing pre-rendered videos (as opposed to performing real-time computation) achieves low latency and high rendering quality even with commodity hardware. There is an added aspect of guided navigation in that the set of navigable videos lay out specific paths of data traversal according to the created roadmap graph. While this limits the total amount of possible exploration, the implication is that a scientist using the authoring tool is familiar with their data. Using his/her domain expertise, the resultant graph is optimized to present the most salient views and tours of the data for educational purposes. This is notable because the application of VR has been shown to be useful in learning domains [34, 39]. Our playback interaction is thus a "choose your own adventure" style of visual storytelling which promotes active learning [35].

Since our approach is especially attractive for affordable HMDs, we design the navigational interface for this context. Our target HMD is Google Cardboard, a budget unit which uses a paired smartphone's gyroscope and mobile GPU to perform head-tracking and scene rendering (specifically, we use a Samsung Galaxy S6). Compared to a comprehensive VR setup with an expensive (and fragile) headset and a high-end desktop with GPU for rendering, Google Cardboard is more accessible in real-world scenarios such as classrooms and museums.

We conduct several case studies and a usability study on our two tools. Despite being prototype-stage software, initial results show the benefits and broader implications of this approach to scientific storytelling. Specifically, they demonstrate the feasibility for presenting high-quality, immersive, stereoscopic, interactive visualizations using affordable and widely-available HMDs.

2 BACKGROUND AND RELATED WORK

Prior work in related areas can be discretized to three main categories: (1) interactive videos, (2) immersive scientific visualization, (3) and animation in the use of visual storytelling. In particular, we address the problem of the lack of effective presentation-based approaches for scientists or educators that want to share datasets on low-cost, immersive, stereoscopic viewing displays.

2.1 Interactive Video

In our context, *interactive* refers to two specific system actions: (1) providing users discrete navigational control over the selection and playthrough of the roadmap of video paths, and (2) 360° viewing that is supported by HMD head tracking. However, what constitutes *interactive video* is more ambiguously defined [31]. Generally, it is premised on providing a flexible, user-directed viewing experience: non-linear playback [32], detail-on-demand video summaries [43], and use of bidirectional or panoramic videos where users can change the viewing direction during playback. Having recognized the benefits of immersion, work has been done to facilitate the production and viewing of immersive, interactive videos [3].

The authoring tool we develop exports video content that is similar to that of Stone et al.'s [45]. They visualize movies of molecular dynamic simulations involving millions of 3D atoms. By incorporating omnidirectional, panoramic techniques in their rendering engine, the resulting movies can be viewed with HMDs. Like them, we use panoramic projection techniques to create immersive and stereoscopic videos (see Section 3.1).

2.2 Immersive Scientific Visualization

Scientific visualization can already be presented on a variety of immersive displays: spherical [5], large-tiled [36], fish-tank [8], CAVEs [50], and HMDs [30]. For HMDs, Drouhard et al. [9] propose design strategies for immersive virtual environments to facilitate the adoption of VR into scientific domains, specifically using the Oculus Rift. They note the potential of consumer-level HMDs to influence the scientific community, with one of its key benefits being affordability. Similarly, our system facilitates knowledge sharing in commodity hardware, but goes even further by focusing on "bargain" units like Google Cardboard. This makes our approach suitable for environments where funding and space are too limited to obtain high-end displays [38].

To provide a comfortable VR experience for complex scientific data, various optimization techniques have been developed to improve the viewing and interaction experience. Targeting non-VR experts, Germans et al. [12] created VIRPI, a high-level, cross-platform toolkit which leverages CAVElibrary for interactive visualization of remote simulations. Ebert et al. [10] use a glyph-based volume renderer which they prefer over isosurface or voxel-based techniques to provide fast rendering times for a stereoscopic system. Kniss et al. [22] implemented a texture-based rendering system for terabyte-sized volume datasets on a high-IO, multi-hardware system. Although this latter example only achieves 5-10 rendered frames per second, their technique provides low latency by modifying the pipeline workload, either by rendering small-but-multiple portions of the display or by performing data subsampling to render fewer samples per frame. In recent work, Hanel et al. [18] continuously adjust the visual quality to serve stable frame rates and prevent simulation sickness.

With navigable videos, we avoid having to use these types of optimizations. Instead, using an advanced renderer in the authoring stage allows us to take advantage of features such as a powerful desktop-based graphics card and advanced lighting, shading, and scene-rendering algorithms; the videos can be exported with the highest settings. As an additional bonus, since the exported video size is small and loaded directly onto the HMD's smartphone, issues like data buffering and network dependencies are not a factor.

2.3 Visual Storytelling and Animation

Navigational videos are designed to support effective content presentation and are inspired by data storytelling and narrative visualization principles [41]. The benefits of storytelling with scientific visualization have previously been recognized [29, 47] and have led to established frameworks for effective communication to target audiences. In practice however, guidelines can be difficult to implement. Gershon and Page succinctly note that, "*a story is worth a thousand pictures*," meaning a single static image cannot capture all the multifaceted components of a story [13]. Fortunately, animation is an effective tool for augmenting storytelling visualizations, but must be appropriately applied in a way that improves the user's experience and visual discourse [7, 19, 37], such as showing causal relationships of objects or events via transitions.

Nowadays, most scientific toolkits include basic animation support and the ability to export to video. More comprehensive systems [4, 27] allow changes for a variety of dimensions by interpolating camera viewpoint, color mapping, and clipping planes. Hsu et al. [20] generate animations by automating camera paths between user-specified keyframes, while Liao et al. [26] use a scientist's interaction history (i.e., provenance).

Zhang et al. [49] note that interactive (and non-linear) videos enhance learner-content interactivity which potentially improves learning effectiveness and motivation. Our authoring tools produces such "stories" around the scientific data by generating interconnected, immersive videos. Similar to [48] and [15], we structure these as a node-link diagram that allows for non-linear animation and navigation. However, our playback interface is a stand-alone application intended for HMDs and other immersive environments, while the renderer/authoring tool is run from a desktop. Instead of allowing completely freeform viewer manipulation of the presentation, we constrain the navigation to the dataset's roadmap of navigable videos. Having the ability to truly explore can verify a user's understanding [17], but alternatively an advantage of our interaction design is that it focuses the aspects of the data that users view only to what the author wants. From a purely narrative perspective, allowing users to deviate from the intended storyline may distract them [44]. Ostensibly, the roadmap author has high insight into the dataset; the created content thus emphasizes the key aspects and features that a viewer should experience.

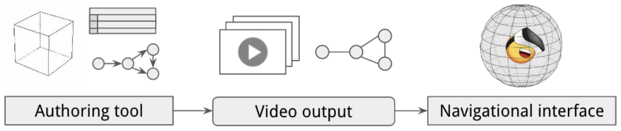


Figure 1: An overview of our system components. The authoring tool creates a graph of interconnected videos. The exported video output (and roadmap graph) are loaded into the navigational interface which allows for traversal and playback.

3 SYSTEM OVERVIEW

Figure 1 shows the steps in taking a raw dataset and converting it to a suitable format for HMD viewing. The authoring tool enables a domain scientist to load a dataset and construct a roadmap graph of animation sequences that traverse throughout the dataset while terminating at shared endpoints. These are exported as a set of high-quality, stereoscopic, 360° output videos and metadata file. The video output is loaded into an HMD, where the navigational interface facilitates immersive, stereoscopic viewing and navigation of the videos.

Since the scope of our work mainly targets HMDs that use smartphones, the navigational interface's user interactions are meant to work by interacting only with the available hardware. The playback tool is built using Unity so it can easily be extended to other VR setups (using their respective SDKs), including ones that can make use of more elaborate hardware components like keyboards, joysticks, paddles, or other paraphernalia.

3.1 Authoring Tool

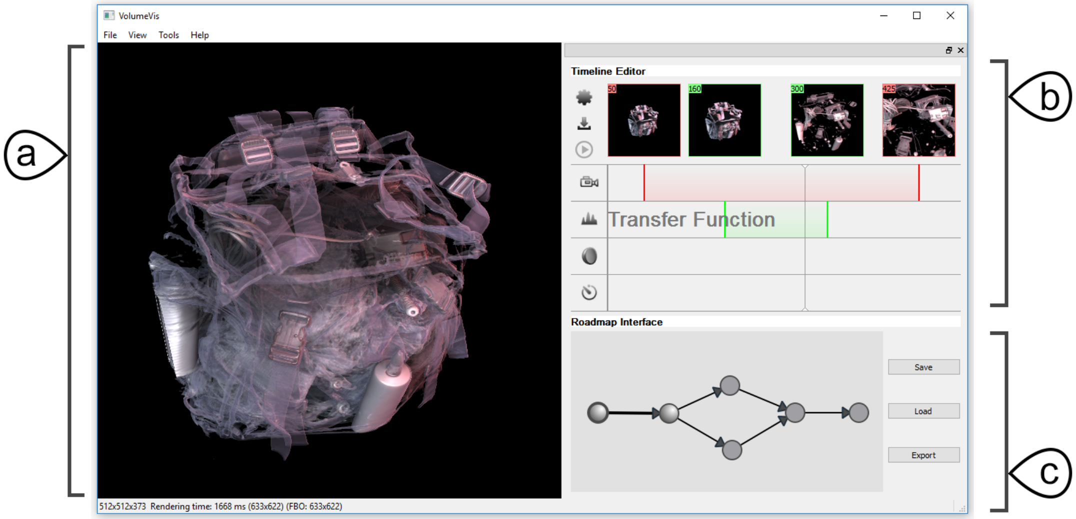
The workflow starts with a domain author loading and visualizing his/her dataset of interest. The authoring stage is where the domain scientist creates a virtual "tour" of the data by building a roadmap. The authoring tool is designed as a modular animation library that can be integrated into an existing desktop-based renderer, allowing flexibility for systems tuned to different types of datasets such as 3D meshes or non-uniform grid data. It takes the form of two panels: a *timeline editor* and a *roadmap interface* (see Figure 2). In our case, we integrate the tool with a renderer for visualizing volumetric data that is uniformly structured on a grid.

3.1.1 Renderer Implementation Details. Using a fast, interactive renderer with a high-end graphics card allows a user to quickly explore the data and experiment with various display parameters, such as color scheme, viewpoint, location, lighting, and transfer function settings. The authoring tool is designed to be renderer-agnostic as it is built using C++, Qt, and FFMPEG for video export, and can thus be integrated into to an existing viewer. (It is connected using Qt's event framework.)

For the volume renderer, a number of minor modifications were made to accommodate rendering in the stereoscopic and omnidirectional manner required by the navigable videos. To do this, we implemented Google's Omni-directional Stereo (ODS) camera model. ODS achieves stereoscopic viewing by producing two panoramic images– one for each eye. We favor this technique as it does not require the composition of sub-images to recreate the projection effects [14].

Additionally, we modified our renderer's ray casting algorithm such that the ray directions match those described in the ODS developer guide [16]. As recommended by the document, we define an interpupillary distance (IPD) of 6.4 cm which is converted to match the units used in the renderer. For advanced rendering techniques, pre-integration alleviates sampling artifacts and volumetric shadows to improve depth cues. We also add the ability to change the clipping plane distance to prevent volume data features from being rendered uncomfortably close to the viewer.

3.1.2 Timeline Editor. Figure 2(b) shows the timeline editor, a keyframe-based panel for creating animation sequences. Several timelines are stacked, each for showing a particular type of action that can be applied for the current animation sequence. For example, the top line shows camera modifications (note the camera icon). In the figure, the camera timeline has a red bar denoting that the user has moved the camera's position between two keyframes; interpolation



512x512x373 Rendering time: 1668 ms (633x622) (FBO: 633x622)

Figure 2: The authoring tool with a sample dataset. (a) The main rendering window shows the current view. (b) The timeline editor is used to created keyframe-based animation sequences; in this case both the camera position and transfer function settings are being changed. (c) The roadmap interface shows a graph of the connected animation sequences. The thicker black edge and highlighted nodes denote the currently selected sequence.

computes the transition. An animation sequence is thus composed of one or more actions (in the figure, a transfer function action is also applied). The timeline editor currently supports four types of actions; each has its own timeline in the panel:

- **Camera** movements change the viewpoint position and viewing angle, either by rotating the camera, performing a fly-through of the data, or panning across it. While the latter two options involve moving the camera's position and are common interactions for presenting volume datasets, camera rotation is a redundant action for HMDs with head tracking support and omnidirectional rendering, as users can simply move their heads to look around.
- **Transfer function** adjustments modify the color mappings and/or opacities for rendered attributes in the data. A user might manipulate this to isolate or highlight particular features of interest that s/he particularly wishes a viewer to see.
- **Clipping planes** are XYZ planes, the positions of which define an axis-aligned bounding box. Values falling outside of the box are removed or clipped. Clipping planes can be used to help reveal the internal structures of the data. Notably, these are often used in medical data visualization as the human body contains many internal structures that must be studied (as in the case study in Section 4.2).
- **Time** refers to the currently shown timestep. For time-varying data, interesting data patterns, trends, or evolutions may reveal themselves over several timesteps.

At the top of the timeline editor, thumbnail snapshots show a visual preview at each created keyframe. Color swatches denote the type of action(s) that each keyframe pertains to. The user can scrub along the timelines to play the animation sequence in the main render window and then update the animation sequence with any desired changes. We also note (as future work) that the editor can be extended to support more general storytelling and presentation actions, such as addition of text annotations and transition effects.

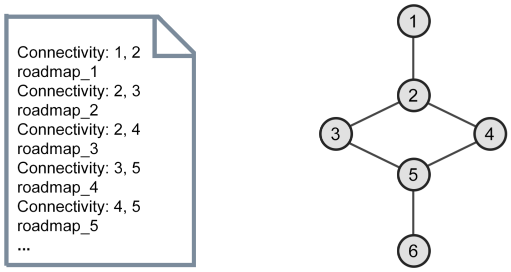
3.1.3 Roadmap Interface. The roadmap interface organizes created animation sequences together into a directed graph. Each edge is a single animation sequence while nodes denote shared keyframes (same camera, transfer function, clipping plane, and time settings) at the terminus of two or more sequences. Edge direction is based on the starting and ending keyframes of its animation sequence. Although viewing of the exported navigable video is bidirectional (you can go forwards or backwards), directed edges are used here to provide a reference to how sequences have been created and to intuitively chain successive ones together.

When building a roadmap, the user can either insert standalone keyframe nodes and subsequently connect them with edges, construct and add new edges organically, or use a mixture of both methods. If a node already exists, creating an exit edge is a branching operation. Alternatively, two edges that have previously branched can be merged together. A new node is created based on the shared ending keyframe settings of the merging edges.

Clicking an existing edge highlights it in the graph and loads its animation sequence into the timeline editor, where it can be reviewed, modified, or deleted. When the user is satisfied with the built roadmap, it can be exported.

3.2 Video Output

Exporting the roadmap from the authoring tool renders a set of immersive, stereoscopic, bidirectional videos, one for each created edge. Each video is encoded using the H.264 codec with FFMPEG, which is supported for encoding and decoding on all Android 3.0+ smartphones [1]. The graph structure is also exported as a metadata



```
Connectivity: 1, 2
roadmap_1
Connectivity: 2, 3
roadmap_2
Connectivity: 2, 4
roadmap_3
Connectivity: 3, 5
roadmap_4
Connectivity: 4, 5
roadmap_5
...
```

Figure 3: (Left) Video output includes a metadata file which lists the graph structure composed of navigable video edges and their connections to endpoint nodes, (Right) used to build the corresponding roadmap.

file which is used by the navigational interface to connect the videos together and facilitate navigation, see Figure 3.

Since viewers should be able to traverse videos at their own pace and in both directions, and H.264 codec does not support backwards playback, we must generate two videos for each rendered animation sequence (one in each direction of playback). Since stereoscopy is also essential for the HMD experience, we generate a video for each eye to achieve proper parallax. As a result, each edge in the roadmap has four videos generated for it– forwards and backwards direction, left and right eye.

In our current setup, we encode video files at 30 frames per second (FPS), which is generally recommended for our targeted set of lower-end HMDs [23]. Colloquially, we find this gives a good balance of file size, I/O bandwidth, and latency. This is especially important since playback in Google Cardboard relies entirely on the smartphone and GPU resources are limited. After performing a small benchmark of dual video playback at various resolutions, we found 720p (1280×720 pixels) to be the maximum resolution supported for our 360° video setup that can efficiently be supported using the mobile device's hardware. (Both this and our chosen FPS are also suggested by Android's developer documentation, see [1].) As this is unfortunately a fairly low resolution, we supersample frames which were rendered as 4K images (3840×2160 pixels) to counterbalance potential visual artifacts like aliasing. However, as smartphone hardware and GPU units will likely become more sophisticated over time, we expect to achieve better video playback capabilities in the future.

3.3 Navigational Interface

After export from the authoring tool, the video output (set of videos and metadata file) can be loaded onto a smartphone and subsequently played using the navigational interface. This smartphone app allows a viewer to navigate, select, and play the roadmapped set of navigable videos. Since video navigation is structured according to the roadmap's graph, viewers traverse edges by playing through one of the videos. Selection of a new edge (i.e., video) occurs at a node after the current video has finished playing, at the point two edges intersect.

The intent of the navigational interface's design is to allow for non-linear exploration via sequence selection and traversal. This lets a viewer experience a "choose-your-own-adventure" form of active learning [35].

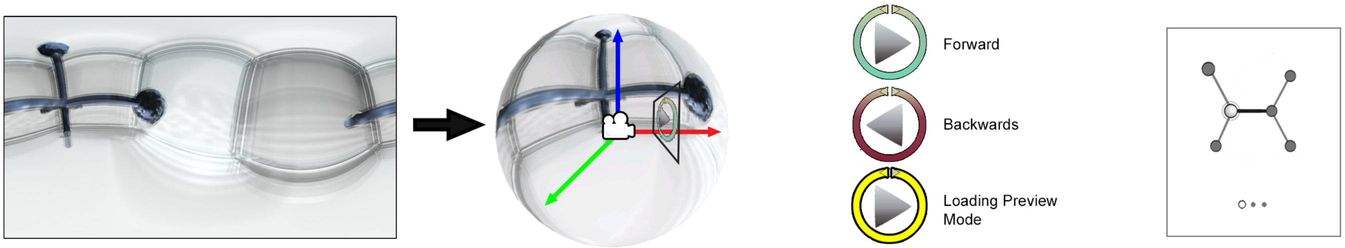
3.3.1 Head-Mounted Displays. Nowadays, several variants of HMDs are available to consumers, such as Google Cardboard, Samsung GearVR, Sony PlayStation VR, Oculus Rift, and HTC Vive. Compared to specialized display systems like CAVEs, each of these headsets provide an affordable alternative. In particular, Google Cardboard presents the cheapest possible platform access, since it is inexpensive and is paired with a user's personal smartphone. The compromise is that viewing quality is reduced compared to higher-end devices like the Oculus Rift or HTC Vive. Still, each of these HMDs features head tracking, stereoscopic viewing, and at least one input element (for the Cardboard, it uses the phone's gyroscope and physical buttons). As previously mentioned, we develop specifically for Google Cardboard due to the potential benefits it can have in widespread educational deployment. However, the navigational interface can also be used on higher-end HMDs and even CAVE systems.

3.3.2 Interface Setup. We use the Unity engine for development as it offers cross-platform support for desktop and mobile deployment along with native VR support and compatibility with several popular HMDs. This allows the navigational interface to run on several different HMDs (and CAVE setups) with little worry about device-specific development (except for device input mappings, which can be expanded to include several inputs in higher-end setups). At development time, Unity did not provide support for video textures on mobile platforms, so we use a third-party plug-in to communicate with the video decoder for rendering the frames to our specified texture.

Figure 4 illustrates how rendering is performed with the navigational interface. A single video frame (in reality two frames, one for each eye) is mapped to a sphere with a two-camera rig placed at its center. To match the authoring tool's interpupillary distance, the rig's two cameras are offset by a 6.4 cm IPD-equivalent in Unity's world coordinates. Rendering a camera for each eye ensures correct parallax. Figure 4(c-d) shows several overlay widgets included in the interface; these are drawn on a plane in world space (versus screen space) to leverage the 3D stereoscopic effects of the view.

When a user begins playing through an animation sequence, we queue the four videos corresponding to that roadmap edge. Only two of the four videos are played at a given time (i.e., the user is either moving forwards, backwards, or playback is temporarily paused). Based on the current play direction, only the appropriate videos are played (the two that play the reverse direction are paused). When the viewer wants to change the playing direction or reaches a video's end, the set of available video paths for the intersection (based on edges connected to the current node) are loaded for syncing.

3.3.3 User Interface Design. The assumption behind the navigational interface's design is that an unobtrusive UI will enhance the exploratory experience. Influenced by Oculus Connect 2's developer conference talk [6], we only utilize a single tactile button for interaction. Based on how a user presses this button, we define the following button actions: *tap*, *double tap*, and *tap + hold.* Within the navigational interface, the viewer is always in one of three user



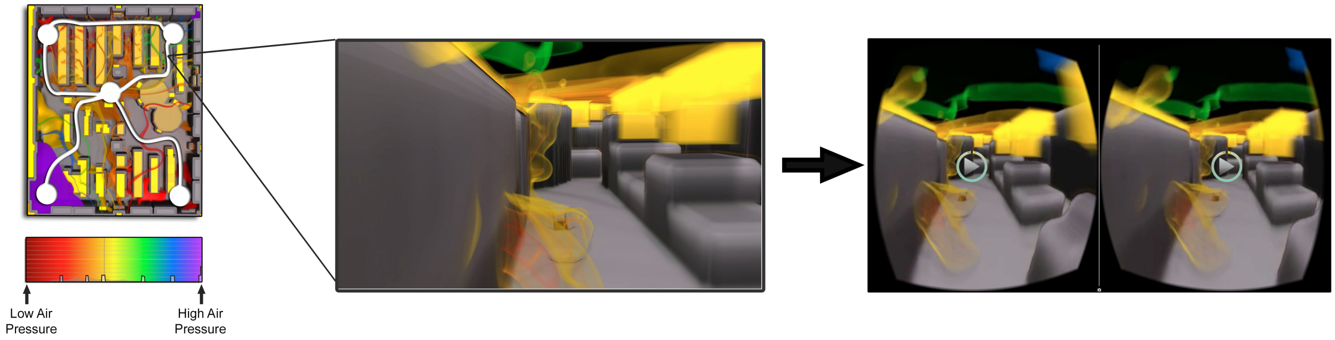
(a) Panoramically rendered image

(b) Rendering to a sphere (with playback widget active)

(c) Playback widgets

(d) Preview mode

Figure 4: Scene setup for the navigational interface. (a) A panoramic video frame is (b) projected onto a sphere with a camera rig at its center. UI widgets are displayed on a plane inside the sphere. (c) Playback widgets show directional progress through video sequences, and (d) the preview mode shows the dataset roadmap and allows sequence selection.



(a) Top-down view of the Server Room with transfer function.

(b) 2D view from inside the Server Room.

(c) The same view rendered stereoscopically for immersive viewing.

Figure 5: Going from raw dataset to stereoscopic view. (a) A top-down view of the Server Room dataset, overlaid with the paths of several camera position animation sequences. The transfer function maps color to air pressure values. (b) A 2D screenshot shows the visualized air pressure gases at a point along one of the animation sequences. (c) The same view rendered stereoscopically.

User State	Button Action	Playback Action
On an edge	Double tap	Switch play direction
	Tap + hold	Play video (release will pause video)
At an intersection	Double tap	Switch play direction (back on edge)
	Tap + hold	Enter preview mode
Preview mode	Tap	Cycle video selection
	Double tap	Exit preview mode
	Tap + hold	Move onto selected video

Table 1: Available interactions in the navigational interface. Depending on the current user state, the three button actions will perform different playback actions.

states: *on an edge*, *at an intersection*, or in *preview mode*. The playback action that the system invokes is defined by the combination of current user state and button action, see Table 1.

When the user is on an edge and thus playing a video, a button tap + hold plays the video forward and moves the user through the animation sequence. Double tapping reverses the play direction (loading the reverse video). When the user is on an edge, a play icon widget shows the current progress of traversing through the animation sequence (Figure 4(c)). At the start of a video, the play icon points right and is surrounded by a circular progress bar. During playback, the bar fills in a counter-clockwise fashion with turquoise. If the user reverses direction and plays the video backwards, the progress bar recedes and the triangle play icon flips to the leftward direction. At any point the user may pause playback (fading out the playback widget) and look around to observe the scene.

Upon reaching a node endpoint (and completing a video traversal), double tapping returns back along the just-completed edge. Holding the button brings up a preview mode: The progress bar turns yellow and the dataset's roadmap fades in (Figure 4(d)). Previously traversed edges are highlighted on the roadmap to give a sense of traversal history. To select a new edge to traverse, the user taps the button to cycle through available animation sequences (based on the current node). Tapping + holding selects the currently cycled edge for playback.

Raw Dataset Details			Authored Content Details				
Name	**Voxel size**	**Total Memory**	**Video Paths**	**Memory Footprint**	**Video Length**	**Avg. FPS**	**Dimensional Changes**
Server Room	417×345×60	0.032 GB	5	0.13 GB	00:01:47	60.401	Camera
Visible Human	Male: 512×512×1877 Female: 512×512×1734	Male: 1.83 GB Female: 1.69GB	7	0.092 GB	00:02:10	59.711	Camera, Transfer Function, Clipping Plane
Supernova (50 timesteps)	867×867×867	120 GB	1	0.54 GB	00:01:40	59.943	Time

Table 2: Details for the case study datasets and their corresponding authored video content. Each dataset is a floating-point, volumetric field using a fixed voxel size.

4 CASE STUDIES

We now present three case studies of turning scientific datasets into navigable videos. Table 2's first three columns note specifications of the raw datasets in the studies; the latter five columns show details for the corresponding authored content. Each case study varies in size and topical domain, they use different animation sequence techniques, one is time-varying, and each presents different features for presentation.

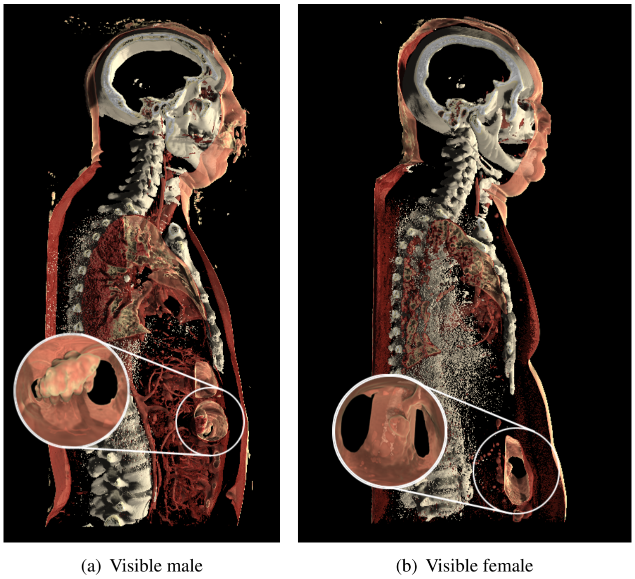
Content was created using the authoring tool on a computer running Windows with a solid state hard drive and Nvidia GeForce GTX TITAN X graphics card. The renderer for the tool employs advanced lighting features to improve the depth perception of the data's features [28]. For HMD presentation, we use Google Cardboard with a Samsung Galaxy S6 containing 32 GB storage, 3 GB RAM, and a display resolution of 2560×1440 pixels.

4.1 Server Room

The first case study comes from a timestep of a simulation of air pressure fields in a ventilated server room. In real-world environments, heat generation from several rows of machine racks can damage expensive computer hardware. Ventilation is thus used to regulate room temperature. The server room is the smallest dataset from the case studies, sized at 417×345×60 voxels and 0.032 GB. To visualize temperature and air flows, we use a transfer function that maps air pressure level to color and opacity (see Figure 5(a)) to highlight both the overall structure and individual flow patterns. The room and machines are colored gray to provide a sharper contrast against the air pressure volumes. By inspecting pockets of air pressure, one can determine where the ventilation can be improved.

After loading the dataset into the authoring tool, we move the camera's position to simulate a first-person view of walking around the room, with a rendered view like in Figure 5(b). Figure 5(a) shows the animation sequences overlaid onto a top-down view of the room. They extend from the center to each of the four corners, as well as along one border.

Using the navigational interface, a viewer "walks" around the room. With 360° viewing and head tracking support, the viewer can stop and look around to see how air usually flows from the floor vents (these areas likely have low pressure) and exhausts towards the ceiling (which have flows of higher green, blue, and purple pressure). Figure 5(c) show a screenshot of the dual stereoscopic rendering that is shown in the HMD. Several thick yellow blocks of air can be seen above machine racks; these come from individual machine exhaust fans which are less powerful than the floor-based ventilation system.


(a) Visible male

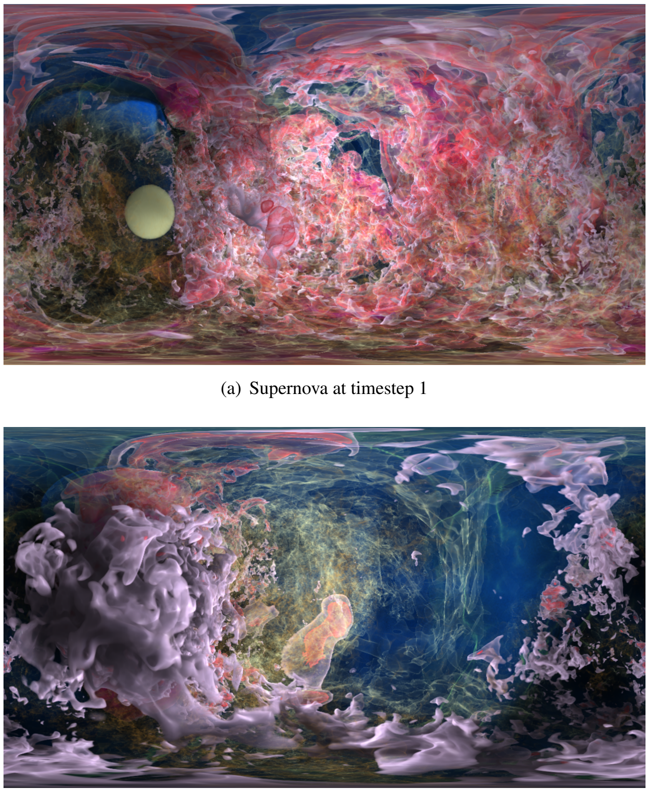
(b) Visible female

Figure 6: The Visible Human case study, where a clipping plane slices open a male and female cadaver to show anatomical differences. For instance, the loupes show two lower body cavities where visual features differ. Navigable paths through these allow for close-up, panoramic comparison.

4.2 Visible Human

The Visible Human dataset is a collection of digitized slices from two full-body cadavers (one male, one female) from the U.S. National Library of Medicine's Visible Human Project [2]. Open datasets like this allow for detailed study of human anatomy. Here, we use the authoring tool to create navigable videos that allow for comparison of anatomical structures between the two bodies.

We create approximately the same roadmap for each dataset. First, we define an initial view of the data (see Figure 6). Animation sequences move the camera by going from the head to the lower chest. These are interspersed with clipping plane adjustments to



(a) Supernova at timestep 1

(b) Supernova at timestep 50

Figure 7: Panoramic renderings from the Supernova case study at the beginning and ending timesteps of the simulation.

reveal internal structures. For the male, additional transfer functions are used to filter out noisy data points in some internal cavities. For the female, camera positions are adjusted slightly to accommodate the different physical dimensions. After exporting the video output, the two sets of video and metadata files were concatenated to a single, bipartite roadmap, allowing the viewer switch between the two datasets at any point in a playback session.

Using the navigational interface, a viewer can explore the physical differences between the male and female. Manipulating the transfer function makes skin values translucent and moving the clipping planes slices into the bodies to reveal organs and bones. Moving the camera position allows comparison of differences at microscopic scales. We annotate Figure 6 with two loupes to demonstrate this. They magnify a section of the lower chest cavity; the male has a protrusion while the female does not. Exploring the differences between the cadavers provides a self-guided learning experience suitable for both medical and biological educational contexts.

We note in reference to Table 2 that this case study has the most dimensional changes and video segments but also the smallest memory footprint. Since camera position doesn't enter inside the cadavers for most of the videos, a sizable portion of output video frames are black. These are highly compressible and result in the small size.

4.3 Supernova

The third case study uses fifty timesteps from a physical model simulation on a supernova. The intent here is to show the supernova's entropy (rate of decline of energy) evolution. These sorts of simulations tend to be large, complex, and multimodal. This makes data processing, analysis, visualization of the raw data challenging. For example, the data might have to be rendered remotely using dedicated servers with the results streamed to a client computer. Even for domain scientists, such setups require dedicated resources and staff; this makes interactive exploration and presentation untenable for other groups (such as introductory astronomy students in a classroom).

The dataset is 867×867×867 voxels; a single timestep is 2.40 GB. The transfer function colors entropy– blue/low to purple/high. Similar to the Server Room case study, the same transfer function settings are maintained for consistent color viewing. The camera position is also fixed in the center of the supernova near its core. We note that datasets like this are normally only viewed from the outside; immersive visualization provides unique views and exploration from within. Only a single animation sequence is generated for this case study, by moving linearly through the fifty timesteps.

Figure 7 shows the supernova's panoramic renderings at timesteps 1 and 50. The end points to the story can be seen with these two views: gas clouds have wrapped around the core (the yellow sphere) and experience high entropy (purple color). Table 2 notes the memory footprint for the raw dataset versus the authored video: 120 GB to 0.54 GB. This small size easily fits into the memory of a smartphone, demonstrating the accessibility of our approach. In contrast, the raw dataset with its large amount of timesteps won't even fit into the memory of most desktop computers, hampering the ability to do interactive rendering.

5 FORMATIVE USABILITY STUDY

To assess the utility of our navigational interface, we conducted a small usability study. This serves as a formative evaluation for feedback on its strengths and weaknesses. Here, we (the authors) assume the role of domain scientists who want to present our work. We use the Server Room dataset (with the same video output from the case study) as our study's dataset.

5.1 Study Procedure

For each subject, a session consisted of four stages: a preliminary assessment, a learning tutorial, a task-based usage stage, and a concluding interview. The preliminary assessment asked about the prior experience each subject had regarding immersive and VR concepts. For the tutorial stage, we verbally explained the goal of this research (*e.g.*, affordable HMD viewing of scientific data) and had each subject go through a short live demo on a sample dataset. We showed them how to use the navigational interface by explaining the system features (the playback widgets, preview mode, and how to interpret the roadmap). If necessary, we also instructed subjects on how to properly hold the Google Cardboard (so they were comfortable and could reach the smartphone button).

For the task-based usage stage, users had to explore the server room by using the navigational interface. They were tasked with answering four questions:

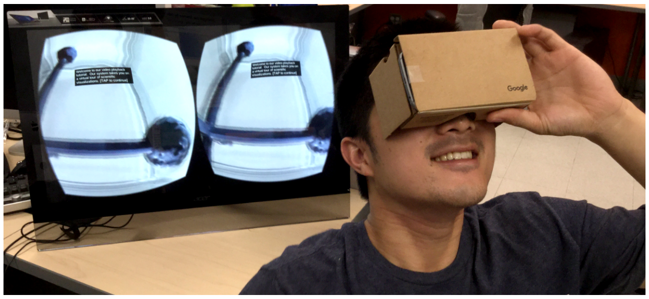


Figure 8: Viewing a sample dataset in the user study.

Q1 Go to two of the room's corners; is the pressure there low or high?
Q2 Are exhaust vents located on the tops or sides of machines?
Q3 Which color(s) emit from the floor vents?
Q4 Does high air pressure exist on the floor or on the tops of machines?

Subjects were allowed to complete tasks in any order and no timing was kept. The tasks encouraged subjects to look around using the 360° navigable videos and to select and traverse different edges in the roadmap. Answers were verbally stated to a moderator, who sat beside each subject and gave assistance when they were confused, stuck, or otherwise needed help.

An interview concluded each session. Here, subjects were asked to rate three system aspects using 5-point Likert scale. They could give freeform commentary and thoughts about what they liked about the playback interface and how it could be improved.

5.2 Participants

Using campus email listservs, we recruited 22 student participants (16 male, mean age=25). All were majors in STEM fields (computer science, biomedical engineering, physics, and material sciences). For the preliminary assessment, subjects reported an average rating of 3.18 (σ=0.89), 3.14 (σ=1.04), and 3.18 (σ=1.05) out of five for familiarity with scientific visualization, VR, and 360° videos, respectively. In the task-based usage stage, two participants failed to complete all tasks, one due to the phone overheating and the other due to experiencing cybersickness.

5.3 Environment

Session setup was consistent for each subject. Participants sat in a swivel chair which allowed them to twist their bodies when viewing in 360°. On a desk besides each subject, we placed paper reference sheets with notes about how the interface widgets worked and summarizing the dataset properties (transfer function settings, etc.). Users were free to remove the HMD from their eyes and review these notes at any time.

We used the same Google Cardboard and Samsung Galaxy S6 setup as the case studies. Google Chromecast streamed the phone's screen to a secondary monitor (an example is shown in Figure 8) to allow the session moderator to see what each subject was currently looking at. Audio was recorded for each session and users were encouraged to take as much time as needed when exploring the content and answering questions.

5.4 User Feedback and Assessment

Subject Likert-scale responses reported an average rating of 3.41 (σ=0.85), 3.86 (σ=0.77), and 3.67 (σ=0.86) out of five for interface usability, presentation effectiveness, and re-usability potential, respectively.

From the concluding interviews we received several comments about the navigational interface. A consistent request was for the roadmap to be always displayed on the view for the stated purpose of allowing quick reference to current location. Several usability features were also proposed. Subject S3 wanted a pop-up help menu: *"maybe if you triple tap [it] show[s] help,"* with formatting similar to that of Table 1. Alternatively, S13 suggested, *"it would be easier if there's a little menu that I can just click [repeatedly instead of several input actions]."* S8 wished for the ability to jump between roadmap nodes: *"I wanted to quickly jump to the corners of the room without playing through the entire video."*

Some subjects related the playback experience to their personal studies. S10 noted, *"I just know this thing in 2D– pressure dispersion around a room. 2D would be as useful to me as 3D. If I didn't know that already, this would be much more useful as a learning experience."* S5 stated, *"I look at proteins on my computer, and so it is very annoying to look on a desktop and just drag it around with a mouse, so I was thinking how nice it is to have 3D."*

6 DISCUSSION AND FUTURE WORK

We now touch on several discussion topics and areas for future work, both for our implemented system and in the broader range of immersive, non-linear storytelling.

6.1 Balancing Author and Viewer Experiences

The case studies and feedback from the user study highlight the tight interplay between the authoring tool and navigational experience. That is, a viewer's experience is a result of the quality of the designed roadmap. There currently are no authoring tool restrictions on how the roadmap can be built. Issues like extreme cases of dimensional change, a large number of short videos, long videos, and non-intuitive animations will negatively impact any playback experience. If the roadmap has "dead ends," the user must play the current video edge backwards to reach an intersection where they can select a new video to play.

Future work can better evaluate user expectations for this sort of HMD-based interaction and propose novel guidelines and strategies for authored content; prior work in (2D) non-linear storytelling can augment this process. When doing this, it will be important to balance the user's experience with the ease and flexibility of authoring content– viewers should feel engaged and in control, while authors should be able to create content that they feel successfully captures their presentation desires.

An important aspect of the viewer experience is minimizing cybersickness. While this wasn't a large factor in our study, it is a common usability issue for HMDs [42]. Navigational interface features such as adjusting the video playback speed based on scene criteria or using easing to transition between animation sequences can be investigated to ensure a smoother experience for viewers.

Additionally, visual cues can be implemented as a way to better emphasize desired storytelling outcomes. A major challenge in immersive, 360° viewing is prompting what a viewer should look at. Since viewing direction is fully guided by head tracking, a user is more prone to missing salient information along a video edge or at a node. We believe that on-screen clues such as text annotations and audio cues can be incorporated as part of the authoring and viewing process to address this issue. However, bidirectional video playback makes this a non-trivial problem since audio and reading is naturally a linear process.

6.2 Improving the Navigational Interface

Qualitatively, there was a fairly high learning curve for using the playback tool, as several users experienced trouble viewing the 360° videos. Specifically, disorientation commonly occurred when camera movements did not align with current viewing direction (*i.e*, when a camera trajectory curved from a linearly forward path while the user kept the same head position). We also noticed participants tended to passively view the content, instead of taking advantage of the 360° viewing by moving their heads around. Despite this, most participants still expressed excitement and piqued interest towards the future applications that our system enables; this is perhaps a byproduct of using a "novel" and "fun" VR gadget.

We note two drawbacks to the current implementation of the roadmap widget. First, it gives no affordances as to what type of animation sequence each edge entails (moving the camera, adjusting the transfer function, etc.). Study subjects inferred a strong connection between roadmap edges and camera movement actions (even during the study's tutorial stage which used a transfer function change). Visual cues or other marks and channels can be used to intuitively indicate the types of dimensional changes that a roadmap edge contains.

The second drawback is that the displayed roadmap gives no reference to the viewer's current spatial position inside the dataset, which can make navigation confusing. Initially this was not so, as we projected animated small multiple previews of available paths based on the current node (instead of showing the roadmap). During a pilot study, subjects complained this feature's implementation was confusingly done since there was no context to current position in the roadmap itself. For the user study, we thus removed the small multiple-based preview mode and instead use the roadmap-based widget described in Section 3.3.2. Based on study feedback, we believe the roadmap widget should include an aspect of spatial position as this is a necessary orientation component for movement, especially at intersections with multiple path choices. We plan to integrate this in future iterations of our system.

6.3 Lessening the HMD Hardware Load

Since we use a third-party plug-in to render frames to a texture, we have less flexibility for memory optimizations of the authored video output. Implementing a custom video encoder and decoder can give better tailored performance, thus lessening the memory footprint on the HMD's smartphone. For example, Facebook's pyramid encoding [24] can be used, which has reported to reduce memory usage up to 85 percent. This would also give better freedom if we choose to render to codecs other than H.264 that are supported on smartphones.

6.4 Immersive Viewing in Other Domains

Our system's design is flexible and can easily be ported to other settings that would benefit from visualization, animation, and the 3D space that VR offers. For example, information visualization is traditionally rendered in 2D and often uses simplification techniques where a large number of data points or values are aggregated, filtered, clustered, or binned together due to the data's density and a lack of rendering and viewing real-estate [40]. Immersive, stereoscopic viewing may be a viable solution to exploring these datasets by using 3D layouts to facilitate level-of-detail zooming and interaction and to provide additional dimensions for display. Dimensional changes in this context can include temporal, motion, color, zoom level, and filtering changes, along with adjusting parameters for clustering, layout, annotation, and aggregation.

In addition to this, we are interested in integrating our system into domain-specific workflows of computational and applied scientists. This would allow for empirical evaluation of the authoring tool and would allow scientists a new medium with which to present findings. Since this population is composed of non-experts when it comes to authoring content and being "data storytellers" (as most educationally-oriented scientific animations are created by skilled animators), it would be helpful to assess how the authoring tool is used "in the field" and thus tailor future iterations of it for their specific usage needs and patterns.

7 CONCLUSIONS

Many scientific studies are about capturing and understanding complex physical phenomena and structures. Immersive visualization is a perceptually effective way to examine and present 3D structures and spatial relationships. To overcome challenges inherent to scientific data, such as large size and complex dimensionality, we develop both an authoring tool and a playback tool. The advantage of our system is that complex data can be fashioned into a non-linear, storytelling-oriented presentation which can then be loaded onto affordable and accessible HMDs. This lowers the bar for showcasing a scientist's research to a target audience and helps open the door for high-quality, immersive education in classrooms where equipping several full-system VR headsets are cost- or space-prohibitive.

Case studies and a formative usability study suggest the promise of our approach. However, further design iterations are needed to improve usability, especially with regards to the navigational interface experience. This is especially true if target audiences include those with relatively little experience (or, frankly, interest) in scientific visualization. We also believe our approach is applicable in other, non-scientific data domains as a novel and storytelling-oriented way to present data.

8 ACKNOWLEDGMENTS

This research is sponsored in part by the National Science Foundation via grants IIS-1528203 and DRL-1323214.

REFERENCES

[1] Supported Media Formats — Android Devleopers. https://developer.android.com/guide/topics/media/media-formats.html. (2016).

[2] Michael J Ackerman. 1998. The visible human project. *Proc. IEEE* 86, 3 (1998), 504–511.

[3] Aseem Agarwala, Ke Colin Zheng, Chris Pal, Maneesh Agrawala, Michael Cohen, Brian Curless, David Salesin, and Richard Szeliski. 2005. Panoramic video textures. In *ACM Transactions on Graphics (TOG)*, Vol. 24. 821–827.

[4] Hiroshi Akiba, Chaoli Wang, and Kwan-Liu Ma. 2010. AniViz: A template-based animation tool for volume visualization. *Computer Graphics and Applications, IEEE* 30, 5 (2010), 61–71.

[5] Xavier Amatriain, JoAnn Kuchera-Morin, Tobias Hollerer, and Stephen Travis Pope. 2009. The allosphere: Immersive multimedia for scientific discovery and artistic exploration. *IEEE MultiMedia* 16, 2 (2009), 0064–75.

[6] Kristoffer Brady and Richard Emms. 2015. Oculus Connect 2:Navigating New Worlds: Designing UI and UX in VR. https://www.youtube.com/watch?v=braV_c4M8oI. (24 September 2015).

[7] Fanny Chevalier, Nathalie Henry Riche, Catherine Plaisant, Amira Chalbi, and Christophe Hurter. 2016. Animations 25 Years Later: New Roles and Opportunities. (2016).

[8] Cagatay Demiralp, Cullen D Jackson, David B Karelitz, Song Zhang, and David H Laidlaw. 2006. Cave and fishtank virtual-reality displays: A qualitative and quantitative comparison. *IEEE transactions on visualization and computer graphics* 12, 3 (2006), 323–330.

[9] Margaret Drouhard, Chad A Steed, Steven Hahn, Thomas Proffen, Jamison Daniel, and Michael Matheson. 2015. Immersive visualization for materials science data analysis using the Oculus Rift. In *Big Data (Big Data), 2015 IEEE International Conference on*. 2453–2461.

[10] David S Ebert, Christopher D Shaw, Amen Zwa, and Cindy Starr. 1996. Two-handed interactive stereoscopic visualization. In *Proceedings of the 7th Conference on Visualization'96*. IEEE Computer Society Press, 205–ff.

[11] Thomas A Funkhouser and Carlo H Séquin. 1993. Adaptive display algorithm for interactive frame rates during visualization of complex virtual environments. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. ACM, 247–254.

[12] Desmond Germans, Hans JW Spoelder, Luc Renambot, and Henri E Bal. 2001. VIRPI: a high-level toolkit for interactive scientific visualization in virtual reality. In *Immersive Projection Technology and Virtual Environments 2001*. Springer, 109–120.

[13] Nahum Gershon and Ward Page. 2001. What storytelling can do for information visualization. *Commun. ACM* 44, 8 (2001), 31–37.

[14] Duke Gledhill, Gui Yun Tian, Dave Taylor, and David Clarke. 2003. Panoramic imaginga review. *Computers & Graphics* 27, 3 (2003), 435–445.

[15] Stefan Göbel, Luca Salvatore, and Robert Konrad. 2008. StoryTec: A digital storytelling platform for the authoring and experiencing of interactive and non-linear stories. In *Automated solutions for Cross Media Content and Multi-channel Distribution, 2008. AXMEDIS'08. International Conference on*. Ieee, 103–110.

[16] Google. 2015. Rendering Omnidirectional Stereo Content. https://developers.google.com/cardboard/jump/rendering-ods-content.pdf. (2015).

[17] Samuel Gratzl, Alexander Lex, Nils Gehlenborg, Nicola Cosgrove, and Marc Streit. 2016. From Visual Exploration to Storytelling and Back Again. *bioRxiv* (2016), 049585.

[18] Claudia Hänel, Benjamin Weyers, Bernd Hentschel, and Torsten W Kuhlen. 2016. Visual Quality Adjustment for Volume Rendering in a Head-Tracked Virtual Environment. *IEEE transactions on visualization and computer graphics* 22, 4 (2016), 1472–1481.

[19] Jeffrey Heer and George Robertson. 2007. Animated transitions in statistical data graphics. *IEEE transactions on visualization and computer graphics* 13, 6 (2007), 1240–1247.

[20] Wei-Hsien Hsu, Yubo Zhang, and Kwan-Liu Ma. 2013. A multi-criteria approach to camera motion design for volume data animation. *Visualization and Computer Graphics, IEEE Transactions on* 19, 12 (2013), 2792–2801.

[21] David Kanter. 2015. Graphics Processing Requirements for Enabling Immersive VR. *AMD White Paper* (2015).

[22] Joe Kniss, Patrick McCormick, Allen McPherson, James Ahrens, James Painter, Alan Keahey, and Charles Hansen. 2001. Interactive texture-based volume rendering for large data sets. *IEEE Computer Graphics and Applications* 21, 4 (2001), 52–61.

[23] Nick Kraakman. 2015. The Best Encoding Settings for Your 4K 360 3D VR Videos. http://www.purplepillvr.com/best-encoding-settings-resolution-for-4k-360-3d-vr-videos/. (1 December 2015).

[24] Evgeny Kuzyakov and David Pio. 2016. Next-generation video encoding techniques for 360 video and VR. https://code.facebook.com/posts/1126354007399553/next-generation-video-encoding-techniques-for-360-video-and-vr/. (21 January 2016).

[25] Bireswar Laha, Kriti Sensharma, James D Schiffbauer, and Doug A Bowman. 2012. Effects of immersion on visual analysis of volume data. *Visualization and Computer Graphics, IEEE Transactions on* 18, 4 (2012), 597–606.

[26] Isaac Liao, Wei-Hsien Hsu, and Kwan-Liu Ma. 2014. Storytelling via Navigation: A Novel Approach to Animation for Scientific Visualization. In *Smart Graphics*. Springer, 1–14.

[27] Ajay Limaye. 2012. Drishti: a volume exploration and presentation tool. In *SPIE Optical Engineering+ Applications*. International Society for Optics and Photonics, 85060X–85060X.

[28] Florian Lindemann and Timo Ropinski. 2011. About the influence of illumination models on image comprehension in direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (2011), 1922–1931.

[29] Kwan-Liu Ma, Isaac Liao, Jennifer Frazier, Helwig Hauser, and H-N Kostis. 2012. Scientific storytelling using visualization. *Computer Graphics and Applications, IEEE* 32, 1 (2012), 12–19.

[30] Stefan Marks, Javier E Estevez, and Andy M Connor. 2014. Towards the Holodeck: fully immersive virtual reality visualisation of scientific and engineering data. In *Proceedings of the 29th International Conference on Image and Vision Computing New Zealand*. ACM, 42–47.

[31] Britta Meixner and Harald Kosch. 2012. Interactive non-linear video: definition and XML structure. In *Proceedings of the 2012 ACM symposium on Document engineering*. 49–58.

[32] Britta Meixner, Beate Siegel, Günther Hölbling, Franz Lehner, and Harald Kosch. 2010. SIVA suite: authoring system and player for interactive non-linear videos. In *Proceedings of the 18th ACM international conference on Multimedia*. 1563–1566.

[33] Jose Noguera and J Roberto Jimenez. 2016. Mobile Volume Rendering: Past, Present and Future. (2016).

[34] Michal Ponder, Bruno Herbelin, Tom Molet, Sebastien Schertenlieb, Branislav Ulicny, George Papagiannakis, Nadia Magnenat-Thalmann, and Daniel Thalmann. 2003. Immersive VR decision training: telling interactive stories featuring advanced virtual human simulation technologies. In *Proceedings of the workshop on Virtual environments 2003*. ACM, 97–106.

[35] Michael Prince. 2004. Does active learning work? A review of the research. *Journal of engineering education* 93, 3 (2004), 223–231.

[36] Khairi Reda, Aaron Knoll, Ken-ichi Nomura, Michael E Papka, Andrew E Johnson, and Jason Leigh. 2013. Visualizing large-scale atomistic simulations in ultra-resolution immersive environments.. In *LDAV*. 59–65.

[37] George Robertson, Roland Fernandez, Danyel Fisher, Bongshin Lee, and John Stasko. 2008. Effectiveness of animation in trend visualization. *Visualization and Computer Graphics, IEEE Transactions on* 14, 6 (2008), 1325–1332.

[38] Maria Roussou. 2000. Immersive interactive virtual reality and informal education. In *Proceedings of User Interfaces for All: Interactive Learning Environments for Children*.

[39] Maria Roussou. 2004. Learning by doing and learning through play: an exploration of interactivity in virtual environments for children. *Computers in Entertainment (CIE)* 2, 1 (2004), 10–10.

[40] Hans-Jörg Schulz and Christophe Hurter. 2013. Grooming the hairball-how to tidy up network visualizations?. In *INFOVIS 2013, IEEE Information Visualization Conference*.

[41] Edward Segel and Jeffrey Heer. 2010. Narrative visualization: Telling stories with data. *Visualization and Computer Graphics, IEEE Transactions on* 16, 6 (2010), 1139–1148.

[42] Sarah Sharples, Sue Cobb, Amanda Moody, and John R Wilson. 2008. Virtual reality induced symptoms and effects (VRISE): Comparison of head mounted display (HMD), desktop and projection display systems. *Displays* 29, 2 (2008), 58–69.

[43] Frank Shipman, Andreas Girgensohn, and Lynn Wilcox. 2003. Hyper-Hitchcock: Towards the easy authoring of interactive video. In *Human-Computer Interaction INTERACT*, Vol. 3. 33–40.

[44] Julie Steele and Noah Iliinsky. 2010. *Beautiful visualization: looking at data through the eyes of experts.* " O'Reilly Media, Inc.".

[45] John E Stone, William R Sherman, and Klaus Schulten. 2016. Immersive molecular visualization with omnidirectional stereoscopic ray tracing and remote rendering. In *2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. IEEE, 1048–1057.

[46] Andries Van Dam, Andrew S Forsberg, David H Laidlaw, Joseph J LaViola Jr, and Rosemary M Simpson. 2000. Immersive VR for scientific visualization: A progress report. *Computer Graphics and Applications, IEEE* 20, 6 (2000), 26–52.

[47] Michael Wohlfart. 2006. Story Telling Aspects in Medical Applications. In *Central European Seminar on Computer Graphics*.

[48] Michael Wohlfart and Helwig Hauser. 2007. Story telling for presentation in volume visualization. In *Proceedings of the 9th Joint Eurographics/IEEE VGTC conference on Visualization*. Eurographics Association, 91–98.

[49] Dongsong Zhang, Lina Zhou, Robert O Briggs, and Jay F Nunamaker. 2006. Instructional video in e-learning: Assessing the impact of interactive video on learning effectiveness. *Information & management* 43, 1 (2006), 15–27.

[50] Song Zhang, Cagatay Demiralp, Daniel F Keefe, Marco DaSilva, David H Laidlaw, BD Greenberg, Peter J Basser, Carlo Pierpaoli, Ennio Antonio Chiocca, and Thomas S Deisboeck. 2001. An immersive virtual environment for DT-MRI volume visualization applications: a case study. In *Visualization, 2001. VIS'01. Proceedings*. IEEE, 437–584.