SAND2020-10233R

E3SM Water Cycle Visualization Project Final Report

TEAM: Erika Roesler, Mark Bolstad, Brad Carvey, Timothy M. Shead, Lauren Wheeler, Andy Salinger (Sandia National Laboratories)

Contact: Erika Roesler (elroesl@sandia.gov)

High-resolution images, a Factsheet, and a digital copy of this report are available here: https://acme-climate.atlassian.net/wiki/spaces/VEWC/pages/1501102165/E3SM+Water+Cycle+Visualization+Project+Final+Report

High-resolution images and animations are available here: https://acme-climate.atlassian.net/wiki/spaces/VEWC/overview and here: https://www.youtube.com/playlist?list=PL1bqZ5tZchtb92cPJ08j7m1qX6szYheAS

Funding:

- DOE's Office of Science, Biological and Environmental Research (BER), Earth and Environmental Systems Sciences Division (EESSD), Data Management Program, Scientific Visualization in Support of the E3SM Water Cycle Simulation Campaign Project

Computing Resources:

- Theta Supercomputer at the Argonne Leadership Computing Facility (ANL)
- Cori Supercomputer at the National Energy Research Scientific Computing Center (NERSC)
- Skybridge Supercomputer (SNL)
- Skybridge Supercomputer (SNL)

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. SAND No.

Section 1 - Introduction and Motivation to Visualize Earth's Water Cycle

Motivated by the need to improve visualizations of Earth's complex water cycle, this team embarked on non-trivial tasks to push from traditional methods of viewing data from simulations as static line graphs and contour plots into new realms with multiple dimensions (three spatial dimensions and the time component) viewable at once. To do this, we chose to feature the extremes in the general circulation of the atmosphere because these Earth system elements are short-lived, but impactful events.

We used simulation data produced by the Department of Energy's (DOE) Energy Exascale Earth System Model (E3SM), which is designed to answer DOE science questions using DOE computing facilities. The E3SM project includes three sets of simulation experiments: cryosphere, bio-geochemical cycles, and the water cycle. The water cycle experiment campaign includes a set of simulations designed to understand how the water cycle will change in coming decades under a changing climate.

Two types of E3SM simulations were visualized. The first type was a coupled-model configuration in which all components of the Earth system from the E3SM were active. To generate this data, E3SM was run in its coupled high-resolution configuration with active land, atmosphere, ice, and ocean components on the Theta supercomputer using ~800 nodes. Data from the atmosphere was output every simulated hour, and data from the ocean output every day.

The second type was an atmosphere-only model with a simplified representation of the cloud physics to simulate a supercell thunderstorm. This high-resolution configuration of E3SM uses a fictional small planet with a radius 1/120th the size of the Earth, making the numerical simulation computationally feasible while providing enough detail to resolve a storm. The supercell thunderstorm appears enormous relative to the fictional planet since it covers roughly one-sixth of the available surface area. The simulation produced timesteps spaced at 15 second intervals, with the thunderstorm developing to full strength over two simulated hours. This version of E3SM in development, testing how high resolutions can more explicitly resolve convection and cloud processes, potentially leading to abandonment of sub-grid paramaterizations at critical resolutions.

In this work, new ways have been explored to visualize the multiple components of the simulated Earth system by using tools capable of projecting time-evolving fluids on a spherical coordinate system. This new set of tools includes ParaView, Houdini, Unity, and the new Universal Scene Description (USD) data format. New hardware has also been tested, allowing users to experience Earth systems data in virtual reality. Care was taken throughout the visualization development to document workflows and use free versions of commercial-off-the-shelf and open-source tools. This was done so that the visualization products could be reproduced by new users as new features in the simulations are discovered.

In one year, August 2019 to August 2020, we completed five major deliverables. For the first deliverable, we created a pipeline for atmosphere and ocean cinema-quality animations and stills

(Section 2). Second, we developed a pipeline for the production of still images and animations for the Supercell storm non-hydrostatic simulation (Section 3). Third, we developed virtual reality visualizations of the Supercell storm (Section 4). Fourth, we created a compilation of this work in a cinema-style movie with video tutorials for reproducibility (Section 5). Finally, we summarized our work in an E3SM Floating Points Newsletter, published on 18 August 2020 (Figure 1). This report describes each of these deliverables with content taken and expanded upon from the newsletter.

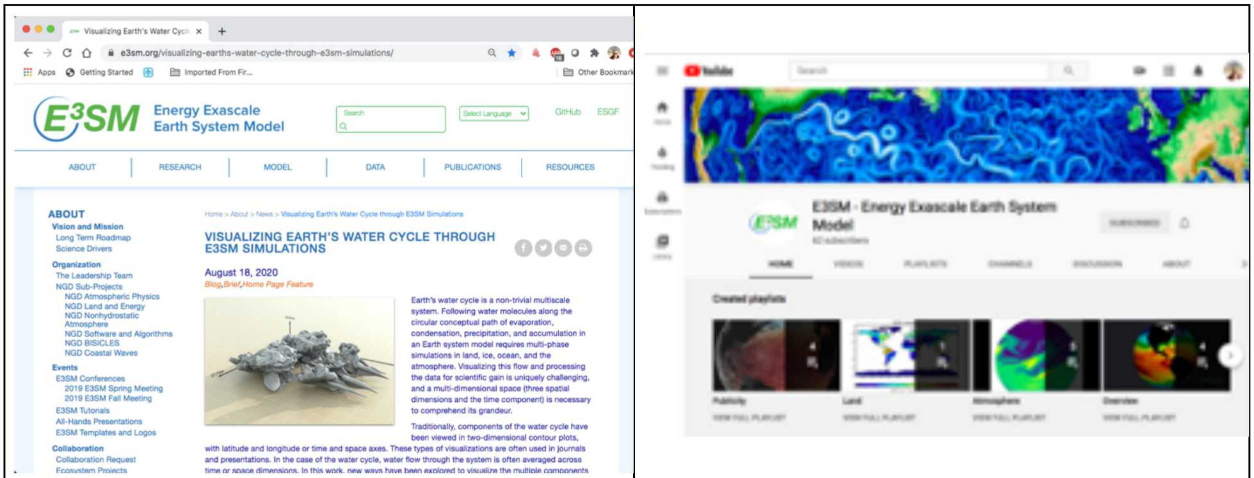


Figure 1. (left) E3SM Floating Points Newsletter with water cycle visualization short story, and (right) the E3SM YouTube channel where many videos from this project are stored. The story can be accessed from https://e3sm.org/visualizing-earths-water-cycle-through-e3sm-simulations/ and the videos from https://www.youtube.com/channel/UCLJ-3mzOcGFn_ZnxNT5IL9Q.

Section 2 - Water Cycle Visualizations

The goal of this task was to develop reproducible and inspiring images and animations from the E3SM Water Cycle simulations using open source software. Early efforts on this task were devoted to initial viewings of the data set and exploring methods of displaying the data to get an understanding of the dataset and available output. This early work was shared with the E3SM community at the 2019 E3SM Fall All Hands Meeting (B12. Earth System Viz, Figure 2). Using Paraview, we explored color maps that are proven to be intuitive and lend themselves to better scientific interpretations (e.g. Wares et al., 2019). We also explored the atmospheric variables associated with the Water Cycle simulations and the different features (e.g., hurricanes, atmospheric rivers, orographic precipitation) that were visible within these fields. This work enabled the team to begin visualizing single time-step and short and low-resolution animations where two-dimensional atmospheric output is vertically scaled, beyond the Earth's radius, to giving the appearance of three dimensions (Figure 3).

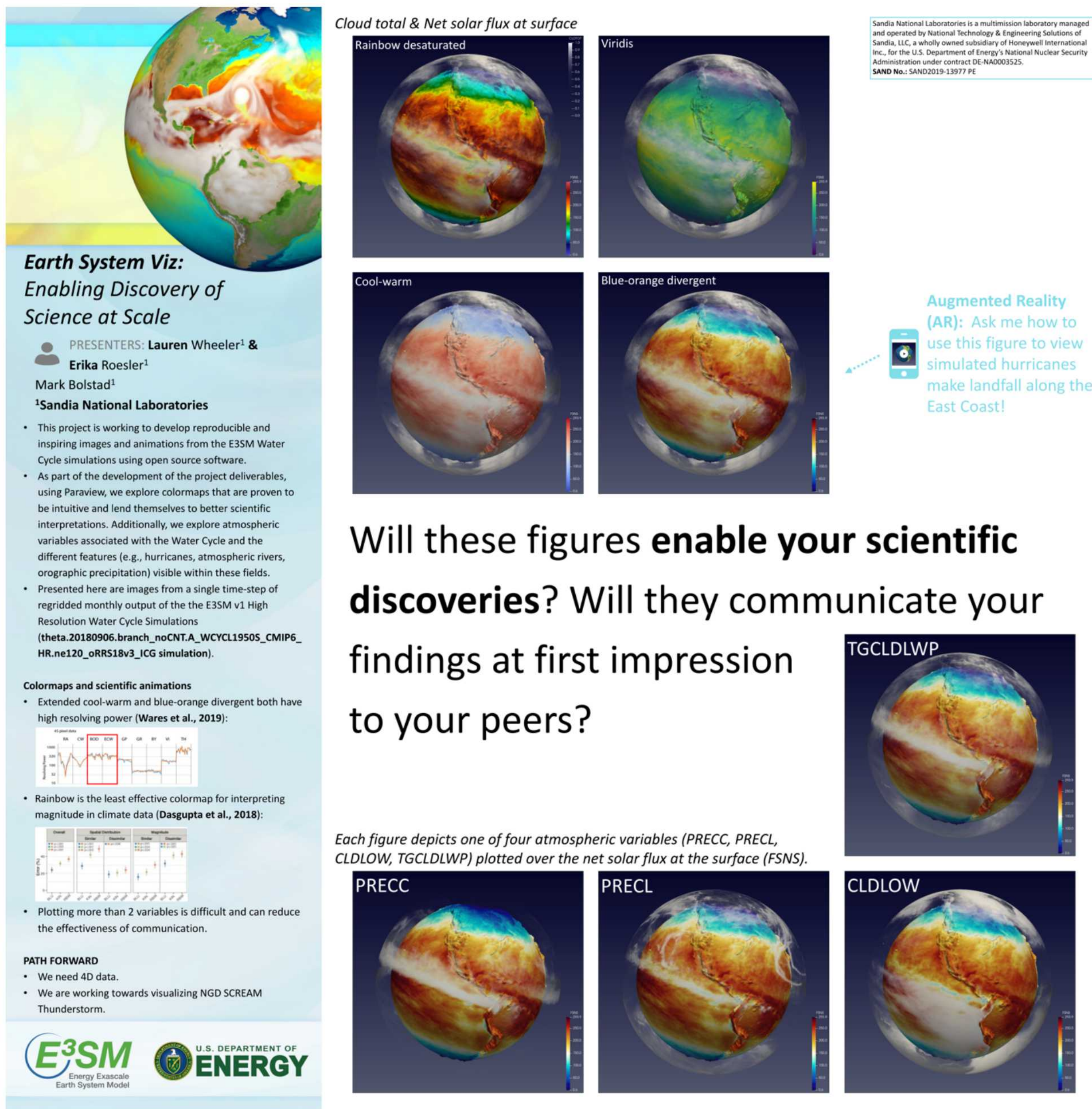


Figure 2. Poster presented during 2019 E3SM Fall All Hands of early work visualizing water cycle simulations

highres_topo_prect_conusview_11152019.mp4

Figure 3. Still (left) and animated (right) of the animation of total precipitation over extruded topography at 9 days. The Paraview state file to create this still and animation is named prect_overlay_CONUSview_highrestopo.pvsm.

The animation in Figure 3 can be reproduced by loading the Paraview state file named prect_overlay_CONUSview_highrestopo.pvsm. Paraview version 5.7.0 was used. To do this:

- Download the following files to a local directory:
    - *prect_overlay_CONUSview_highrestopo.pvsm*
    - *USGS_gtopo30_0.23x0.31_remap_c061107.nc*
    - *theta.20180906.branch_noCNT.A_WCYCL1950S_CMIP6_HR.ne120_oRRS18v3_ICG.cam.h3.0046-08-29-21600.nc*
- Open ParaView and under *File* select *Load State*.
    - Select the state file: *prect_overlay_CONUSview_highrestopo.pvsm*
- In the *Load State Options* dialog window, select *Search files under specified directory* and then select the directory where the following two data files are stored.

An external tutorial for extruding the topography can be found here under the *Spherical representation* section on: https://www.dkrz.de/up/services/analysis/visualization/sw/paraview/tutorial/topography-extrusion. The following equation is from the above tutorial and can be copy and pasted directly into ParaView's Calculator filter. Changing the highlighted value controls the degree of scaling and changing the highlighted variable controls the variable to be scaled.

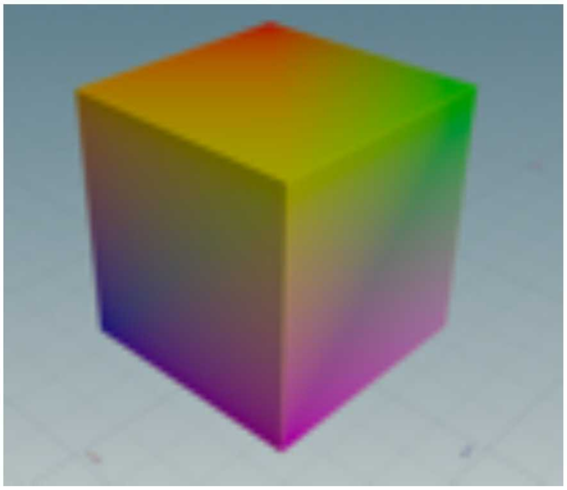
- (1 + (PHIS/6370000) * 35)* (iHat * cos(asin(coordsZ)) * cos(atan(coordsY/coordsX)) * coordsX/abs(coordsX) + jHat * cos(asin(coordsZ)) * sin(atan(coordsY/coordsX)) * coordsX/abs(coordsX) + kHat * coordsZ)

The next step in the coupled water cycle visualization was to view more than one component at a time. One of the major difficulties of this project occurred here, when we realized data from each component was output at different timescales and the necessary cadence of output was not available in the E3SM data archive. We began collaborating with E3SM team members regularly running the coupled model for tuning and testing and asked for a dataset that could be output with a time range appropriate for extreme water cycle visualizations. Once the data was acquired, it was found the high time frequency ocean output from MPAS-O has a bug where the last time steps of the month are not written to the output file. This requires a time smoothing or interpolation between the last available time step from one month and the first available time step from the proceding month to be smoothed together. In this process, we also realized visualizing ocean dynamics has much slower cadences than the atmospheric dynamics. One exception to this discovery are cold wake plumes in the ocean following cyclonic storms. This led to the new development of showing one component first, i.e., the ocean, so that the viewer can absorb that information, followed by adding another component, i.e., the atmosphere, for more information to the viewer.

This process of creating a visualization of multiple components now has a robust workflow pipeline to enable reproducibility for any future E3SM simulation. This pipelines accommodates each component of the dataset outputting at different timescales, ranging from hourly to monthly. SideFX's Houdini software was used for rendering and animation, and Pixar's USD was used for data interchange. Universal Scene Description (USD), released to the open source community in 2016, is a core component of Pixar Animation Studio's graphics and rendering pipeline. An overview of USD(http://graphics.pixar.com/usd/docs/Introduction-to-USD.html) describes its capabilities as:

> "USD provides for interchange of elemental assets (e.g. models) or animations. But unlike other interchange packages, USD also enables assembly and organization of any number of assets into virtual sets, scenes, and shots, transmit them from application to application, and non-destructively edit them (as overrides), with a single, consistent API, in a single scenegraph. USD provides a rich toolset for reading, writing, editing, and rapidly previewing 3D geometry and shading. In addition, because USD's core scenegraph and "composition engine" are agnostic of 3D, USD can be extended in a maintainable way to encode and compose data in other domains."

USD provides an API for creating scenes and geometry, and provides a high-performance OpenGL based application for previewing the generated files. The generated files can be an easy to read and edit in their ASCII format (see figure 4, for an example file and the geometry it describes), and a binary format that is compact and allows for non-linear access. USD has been widely adopted across a wide variety of DCC tools (Maya, Houdini, and Katana with Blender and 3DMax support under development) and other entertainment tools such as Unity and Unreal Engine. As such, it makes an attractive target to bridge the gap between scientific data and tools for cinematic visualization. USD can represent all of the geometry types needed for the final visualization along with the multiple timescales required for the project. A C++ tool was developed that could ingest most of the file formats exportable from ParaView (and by extension, VTK, the underlying framework for ParaView), so the resulting workflow was *E3SM Data → ParaView → C++ tool → USD*.



```
#usda 1.0
(
  defaultPrim = "World"
  endTimeCode = 1
  startTimeCode = 1
  upAxis = "Y"
)
def Xform "World
{
  def Mesh "mesh_0"
  {
    float3[] extent.timeSamples = {
      1: [(-0.5, -0.5, -0.5), (0.5, 0.5, 0.5)],
    }
    int[] faceVertexCounts.timeSamples = {
      1: [4, 4, 4, 4, 4, 4],
    }
    int[] faceVertexIndices.timeSamples = {
      1: [1, 5, 4, 0, 2, 6, 5, 1, 3, 7, 6, 2, 0, 4, 7, 3, 2, 1, 0, 3, 5, 6, 7, 4],
    }
    point3f[] points.timeSamples = {
      1: [(-0.5, -0.5, -0.5), (0.5, -0.5, -0.5), (0.5, -0.5, 0.5), (-0.5, -0.5, 0.5),
          (-0.5, 0.5, -0.5), (0.5, 0.5, -0.5), (0.5, 0.5, 0.5), (-0.5, 0.5, 0.5)],
    }
    color3f[] primvars:displayColor (
      interpolation = "vertex"
    )
    color3f[] primvars:displayColor.timeSamples = {
      1: [(1, 0, 0), (1, 0.5, 0), (1, 1, 0), (0, 1, 0),
          (0, 1, 1), (0, 0, 1), (1, 0, 1), (1, 0.5, 1)],
    }
  }
}
```

Figure 4. Simple ASCII USD file describing a cube with a unique color per vertex (top) and the resulting image (bottom).

For the simulations, the ocean data is output on a two-dimensional, unstructured, time-varying mesh using daily timesteps, and the atmospheric data is output on a three-dimensional, unstructured, time-varying mesh using hourly timesteps. A series of ParaView batch scripts load and then export the E3SM data with a reduced set of variables into parallel ParaView unstructured data files. These files are then converted to the USD format and imported into Houdini for animation and rendering.

Following import the team explored how to handle the different timesteps for the different components. The default for USD is to hold a timestep of data across however many sub-timesteps are in the animation, until the next valid timestep for the data is reached. In the case of the E3SM data, the daily ocean data is held across the 24 sub-steps of the hourly atmospheric data. However, USD and Houdini provide the flexibility to interpolate the ocean data to any frequency or sub-frequency of the atmospheric data. For example, the ocean data can be interpolated to one-hour intervals or six-hour intervals if the user chooses to do so. The result can be seen in Figure 5.

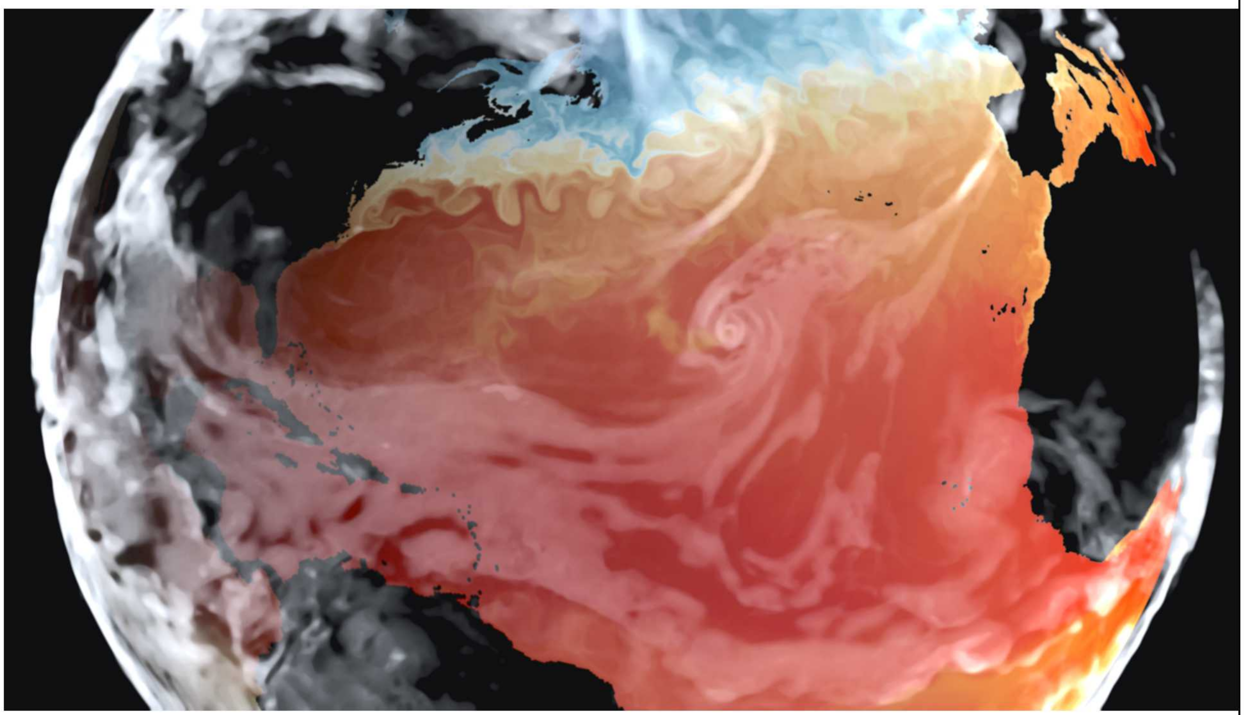


Figure 5. The image is taken from about 1 minute into the animation, also serving as a placeholder to the hard copy version of this report. (Transition from an ocean-only visualization to the coupled atmosphere-ocean visualization. At the beginning of the visualization on the left, the ocean data is on daily time steps. Once the visualization focuses on the Atlantic ocean, the data is slowed down and interpolated to hourly time steps to match the atmospheric data. Throughout the visualizations, lighter colored streaks occur through the ocean data. These are cold water wake created by storms that can be seen in the latter portion of the visualization. The landmasses are black, the ocean is visualized with warmer and cooler colored contours of sea-surface temperatures, and the atmosphere's cloud fraction is in grey-scale. The visualization on the left is the a zoomed in sequence of the atmospheric data on the right, with the lower cloud fraction made more transparent for the wakes to be more visible.)

Compositing was the final step of creating the water cycle visualization. Figure 6 shows a typical decision flow of how cold wakes and cloud layers can obscure each other. Artistic decisions are needed to retain these features.

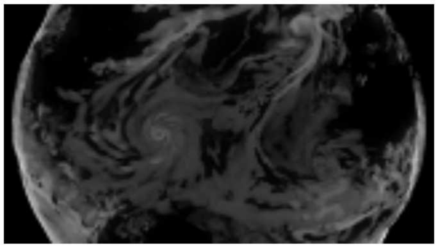
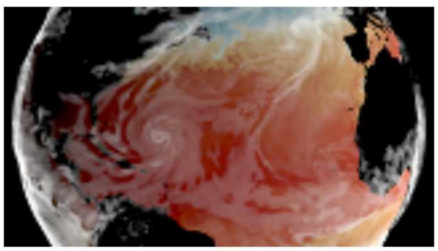
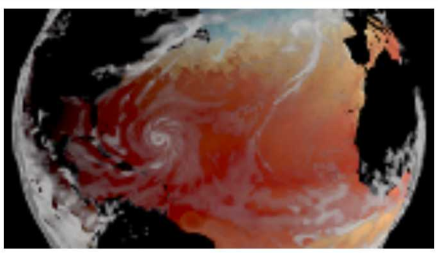
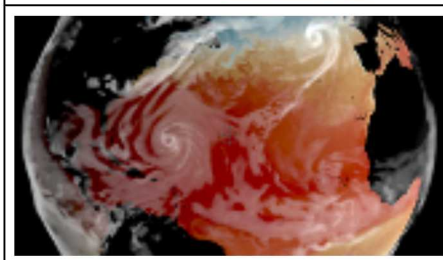
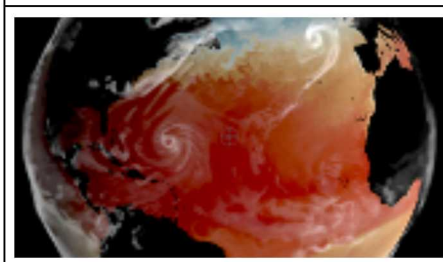
	Atmosphere data out of render without composition.
	Composited atmosphere and ocean data with minimal tweaking.
	Brightening the atmosphere to make it more prevalent, but can potentially hide cold water wakes from hurricanes.
	Removing and reducing the low values to make the hurricanes more prominent, but atmospheric values on edge of globe are altered.
	Increasing the saturation of the colors and lightening the edge of the globe over the Pacific.
	Final. Same as above, but with increase focus on the wakes.

Figure 6. Steps of composition sequence after rendering in Houdini for ocean and atmosphere data for the E3SM Water Cycle simulation experiment.

Section 3 - Supercell Storm Visualizations

SideFX Houdini is a technical artists' tool that is widely used in film special effects and game development, implementing a sophisticated system of hierarchical data flow networks. Users create and connect networks of *operators*, where operators act as data sources, data filters, and data sinks, with data flowing from one operator to the next. Upstream changes to data sources and parameters automatically trigger recomputation of the network to produce new results. New ideas can be tested in parallel branches of the network, and the number of ways in which operators can be combined is limited only by the artist's imagination. This made Houdini ideal as the primary data wrangling and visualization tool for the supercell thunderstorm simulation; due to the flexibility of the network data flow design, we were able to bypass multiple layers of data manipulation needed elsewhere in the project, developing a single unified workflow for all of the supercell visualizations, including a Virtual Reality (VR) visualization described later.

The raw simulation outputs were delivered to us for visualization as a collection of 481 NetCDF files, one per simulation timestep. The files required 20TB of disk space. Each file contained data that had been mapped from the original cubed-grid representation of the simulation to a latitude-longitude grid, as the latter produced higher-quality results in our initial testing. In addition to metadata, the files contained dense arrays, one per simulation field, that were 128 x 1537 x 3072, i.e. they stored values on a 1537 x 3072 latitude-longitude grid, at each of 128 levels in the atmosphere. Simulation fields stored in each file included water vapor, liquid water, and rain density; potential temperature; and latitudinal, longitudinal, and vertical wind components.

Houdini provides builtin *volume* data structures, which are strongly typed dense 3D arrays. These were a perfect match for the 3D field data in our NetCDF files. To ingest the NetCDF data as Houdini volumes, we used Houdini *Python* operators as the inputs to our pipeline. The functionality of these operators is user-defined by code written in the Python programming language, making them particularly flexible tools for data ingestion and manipulation. In our case, the code in our *Python* operators used the Python netCDF4 module to load field arrays directly from the NetCDF files into our pipeline as Houdini volumes. To generate animations, Houdini's "current frame" was used in a parameter expression to specify the NetCDF filenames to be ingested; thus, changes to Houdini's current frame automatically triggered a load of the corresponding data, and recomputation of the downstream network nodes.

Once the raw data was loaded into Houdini volumes, we used a wide variety of operators to combine and/or modify them. For example, we used the *Volume Merge* operator to combine the liquid water and rain volumes to produce a derived volume useful for creating visualizations based on total liquid water. Similarly, because Houdini supports both scalar and vector volume types, we used the *VDB Vector Merge* operator to combine the three separate volumes storing wind velocity components into a single vector-valued volume that was easier to use in downstream operators. In another example, we converted the raw potential temperature volume to real temperature using the *Volume Wrangle* operator, which modifies every cell in a volume using a user-defined expression written in a domain specific language called VEX. Although less general than Python, VEX code runs much faster because it can be executed in parallel using just-in-time compilation, making it ideal for our large field volumes. As a more sophisticated example of derived data, we used several *VDB Analysis* operators to compute volumes containing the 3D velocity gradients for the wind, which were used as inputs to another *Volume Wrangle* operator that computed the Q-Criterion, a popular metric for capturing vorticity. At several points in the network, we used *File Cache* operators to cache derived data to disk, making downstream computation more responsive.

To create isosurface visualizations we fed volumes to the *Convert* operator, which converted them to polygons using scalar values chosen by inspection to define levelset boundaries. In some cases we followed *Convert* with a *Smooth* operator to reduce artifacts introduced by the isosurface extraction. We used *PolyFill* operators to fill holes in the polygonal surface, such as a large hole in the base of the isosurface where where the storm met the simulation / ground boundary. In some cases we used the *Clip* operator to remove half of the storm where it straddled the equator and was roughly symmetric. Splitting the isosurface in half this way introduced more holes that were again filled with *PolyFill*. To display additional information on the clipping plane thus created, we used the *Attribute from Volume* operator to map data contained in a volume to the polygons comprising the clipping plane, and a *Color* operator to map data values to a color ramp. For example, we created visualizations using the total liquid water volume with the real temperature displayed as a color gradient on a clipping plane.

For streamline visualizations we used the *Scatter* operator to generate seed points spread at random throughout the wind volume. We then used the seed points and the wind field as inputs to the *Volume Trail* operator, which advected the seed points through the volume to create polygonal streamlines. At this point in the process the streamlines were simply one-dimensional lines embedded in three-dimensional space. Because the seed points were scattered evenly throughout the volume, so were the streamlines, including many "boring" streamlines that moved through the volume with little interaction with the storm, obscuring the storm within. To automatically eliminate these "boring" streamlines, we used a *Polygon Wrangle* which, like the *Volume Wrangle* described above, applies a VEX expression to every polygon (in our case every streamline) in its input. Here, we used it to compute a "complexity" metric which was the sum of the angles between adjacent streamline segments. Thus, lower complexity values were assigned to streamlines that were "straight", and higher complexity values were assigned to streamlines that were "wiggly". We used a *Group* operator to define a group containing all of the streamlines whose complexity was lower than a threshold, and the *Delete* operator to remove the entire group. This process removed the bulk of the boring streamlines, but we were still left with a few "stragglers" that could be distracting, depending on the camera view. Where necessary we used interactive manual selection with the *Delete* operator to remove them; this was an excellent example of how the flexible data manipulation of an artists' content creation tool, which might otherwise be frowned upon, can have a valuable place in scientific visualization. With the stragglers removed, we used the *Color* operator to assign a color ramp based on wind velocity, and the *PolyWire* operator to convert the one-dimensional streamlines into three-dimensional "tubes" for final rendering.

Another example where Houdini's content creation proved useful was generating schematic dimensional information for the storm. Rather than simply superimposing the storm's dimensions over a rendered image, we wanted the dimensions to be aligned with the storm's boundary in 3D space, and updated automatically as the storm developed. To do this we used the *Bounds* operator to extract the size of the storm, along with the *Add* operator to generate dimension lines for each dimension, followed by the *Sweep* operator to give the dimension lines thickness and arrow heads. *Font* operators were used to convert the length of each dimension line into formatted polygonal text near each arrow head. In early tests generating dimension lines for isosurfaces, we found that the dimensions changed so continuously and erratically that they were difficult to read. Reducing the precision in the dimension values to a single decimal place helped, but was still dissatisfying. To address this, we placed a *Solver* operator downstream from the *Bounds* operator, modifying the computed bounds using an exponential moving average. This made the behavior of the dimension lines much less frantic, dramatically improving their readability, at the cost of introducing a small amount of latency.

Because the data we were loading was based on a flat latitude-longitude grid, so were our visualizations. In some cases, we wanted to place the storm in context by visualizing it using spherical coordinates on the surface of the fictional reduced Earth used for the simulation. To do this, we needed to apply a nonlinear transformation to "bend" the flat latitude-longitude grid, "draping" it over the surface of a sphere. In addition, we wanted to animate the transition between latitude-longitude and spherical coordinates. Furthermore, we wanted to be able to apply this transition to any of our visualizations, including isosurfaces, streamlines, dimension lines, and any other content we might create. This was yet another case where using a full-fledged content creation tool instead of a more static data visualization tool paid off. We used a *Box* operator to produce a polygonal grid with the same topology as that of the original simulation, and a *Point Wrangle* operator to conform the box to a sphere. This geometry was used to represent the surface of the fictional planet.

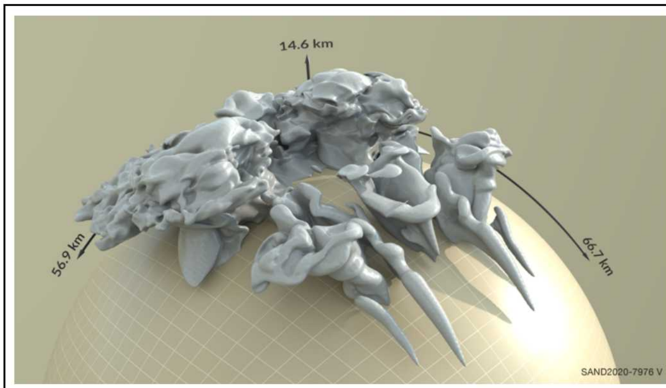
To create the animated projection from latitude-longitude coordinates to spherical coordinates, we used the *Sphere* operator to generate a set of points on the surface of a sphere, and the *UV Project* operator to assign latitude-longitude coordinates to each point. The *Copy and Transform* operator was used to make multiple duplicates of the points, creating a three-dimensional spherical lattice that extended from the surface of the simulated planet to the top of the simulated atmosphere. A *Point Wrangle* operator mapped the latitude-longitude coordinates stored on each point to a coordinate system where latitude, longitude, and altitude mapped directly to X, Y, and Z. Thus, we had two versions of the lattice, "spherical" and "flat". A second *Point Wrangle* was used to interpolate between the two versions of the lattice, using a "mix" parameter that could be animated to produce a lattice in any intermediate configuration between spherical and flat. We used the *Point Deform* operator to deform our polygonal geometry based on the difference between the original and transformed lattice. For example, we used *Point Deform* with both isosurfaces and the fictional planet surface, so that both would deform simultaneously and in step.

In all cases, we used the path tracing Mantra render engine provided by Houdini to render high-quality images and animation using motion blur, ambient occlusion, and environment mapping. Our background sky, planet, and ground colors were chosen from a neutral palette designed to highlight the storm without competing for the viewer's attention. The sky was deliberately given an unnatural color to emphasize the synthetic, fictional-planetary origin of the data. To render the planet and ground, we created a special high quality "wireframe" shader to emphasize the topology of the underlying grid. This was particularly effective when rendering the animated transition from spherical to lat-lon coordinates, as it clearly shows how the spherical grid **does not** align with a lat-long grid. In addition, the featureless, abstract wireframe grid also served to ensure that viewers would not mistake the atmosphere-only simulation for a more sophisticated, coupled model. To render isosurfaces we used a slightly reflective material with many of the qualities of porcelain, as the subtle reflections and highlights helped to emphasize the complex shape and curvature of the levelset. Lighting was provided by an HDRI environment map, yielding soft shadows that also helped to reveal form, without being harsh or distracting.

Although we were pleased with our dynamically generated dimension lines, we did conduct experiments with alternative backgrounds that could provide a more intuitive human understanding of the size and scale of the storm, with the understanding that this was at odds with our previously stated desire to prevent viewers from reading too much into the images. After many experiments we settled on a volume rendering of the storm that had a reasonably cloudlike appearance, against a backdrop of real world photographic data obtained from the National Agriculture Imagery Program, which collects ortho imagery using aerial photography across parts of the United States at up to 30 cm resolution.

A notable challenge in creating the volume rendering was settling on an appropriate cloud density. A literature search showed that, within a thunderstorm of the type simulated, the mean free path of a photon is 10-30 meters between scattering events. As a practical matter, this means that the interior of the cloud is completely opaque beyond 100 meters from the surface. In our case, each cell within the thunderstorm simulation grid was roughly 150 meters on a side. This meant that when we specified physically-accurate values for the cloud density, the transition from 100 percent transparent to 100 percent opaque took less than one cell, which could not be resolved and led to objectionable visual aliasing artifacts. As a compromise, we reduced the cloud density below realistic values to reduce the aliasing, and the resulting image does have the desired effect of emphasizing the size of the storm.

This work is culminated in the series of images in Figure 7, which shows the flat surface, draping over a sphere, streamlines, temperature contours, scale, and projection on aerial photography.



Still image of Supercell on curved surface, small Earth radius, with dimensions.

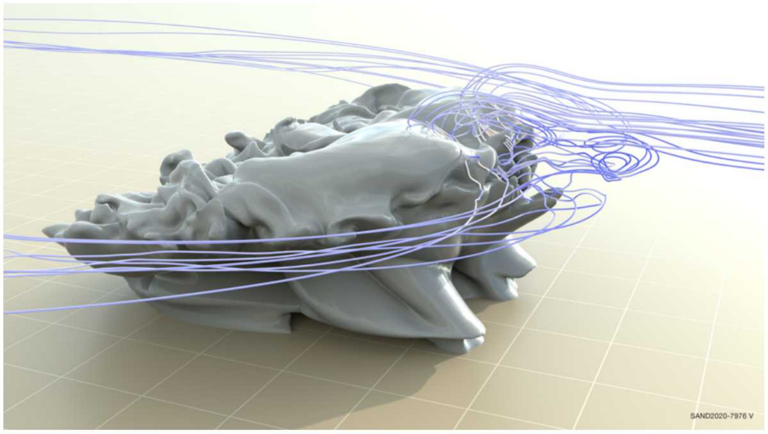
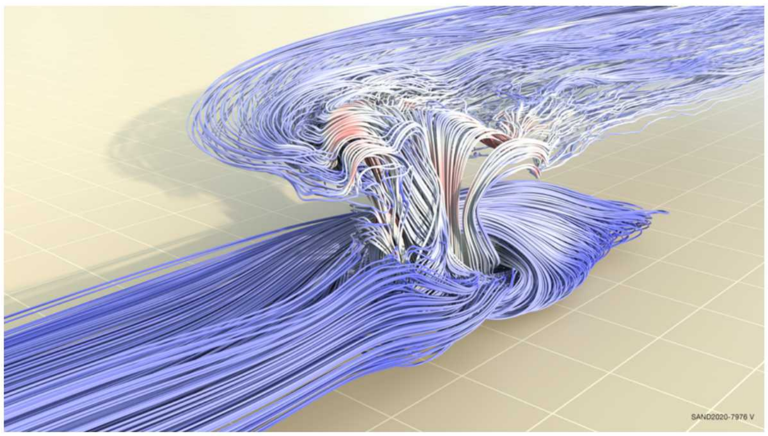
	Still image of supercell on flat surface, streamlines moving around condensed water.
	Still image of Supercell's streamlines showing complexity and strength of updrafts during mature storm.
	Artistic rendering of mature supercell over farmland from small Earth radius simulation.

Figure 7. Sequence of images produced to visualize supercell simulation from the nonhydrostatic small Earth radius. Supercell storm matured over a flat surface, draping over a sphere, streamlines, temperature contours, scale, and projection on aerial photography. Animations of the storm with transition from spherical to lat-lon coordinates, supercell and dimensions as storm matures and progresses during simulation, and temperature mapped to a clipping plane through the center of the storm exist in the digital versions of this report. See header of this document for animation download locations.

Section 4 - Virtual Reality Visualizations

In addition to our more traditional visualizations, we experimented with a virtual reality (VR) visualization of the supercell thunderstorm simulation. The hardware for the VR visualization was wireless HTC Vive Pro headsets with six degrees of freedom, connected to HP Omen-X Gaming Cube workstations with nVidia 1080 TI GPUs. Our software stack used the SteamVR API running in a custom application developed using the Unity gaming platform. As mentioned above, Houdini is widely used in the gaming industry, making it an excellent choice to convert the thunderstorm data to representations and file formats that were usable in Unity.

To ensure user comfort in VR, we needed our application to maintain a stereo frame refresh rate of 90 Hz. For this reason our initial focus was on converting the raw simulation data into polygonal isosurfaces that could be rendered quickly by Unity's realtime render engine. We added a set of branches to the Houdini network pipeline described previously, which were focused on generating data for VR. These branches took the field volume data and converted it to isosurfaces using a lower level of detail than our earlier visualizations. This was done with two fields - liquid water and rain - and the results were saved using the widely-supported Wavefront OBJ file format, storing one file per field, per timestep. This required 11 GB of storage. To give users a sense of the scale of the storm, we substituted a 3D topological mesh depicting a hundred mile stretch of the Rio Grande valley where Sandia National Laboratories is located. The mesh is centered on the Sandia and Manzano mountain ranges which are widely recognized landmarks for Sandians, and roughly the same length as the storm.

Within our Unity application, the individual timestep files are all loaded at program startup, and users are placed in a virtual environment based on a field outdoors. We choose to provide a pleasant backdrop because we find that a formless void tends to cause user anxiety. Each VR user holds a pair of controllers in their hands, and the controllers are tracked and visible in the VR environment. Using buttons and trackpads on the controllers, users can display and interact with a virtual menu system in VR. For example, they can choose which fields to view, and control playback of the simulation using a standard VCR button metaphor to play, pause, fast forward, or rewind the simulation. In addition, users can move, rotate, and pinch-to-zoom the simulated storm using their controllers to view the data from any viewpoint and at any reasonable size.

Thanks to our prior VR work on other projects, our application includes sophisticated collaboration features, so that multiple VR users can be networked into the same virtual environment, seeing each other's avatars and synchronizing state changes so that they view and interact with the same data at the same time. In addition, we provide an "observer mode", allowing users who don't have VR hardware to run the application and participate in a collaborative session using a standard mouse and keyboard interface.

Figure 8 shows stills of the Virtual Reality experience of the Supercell storm.

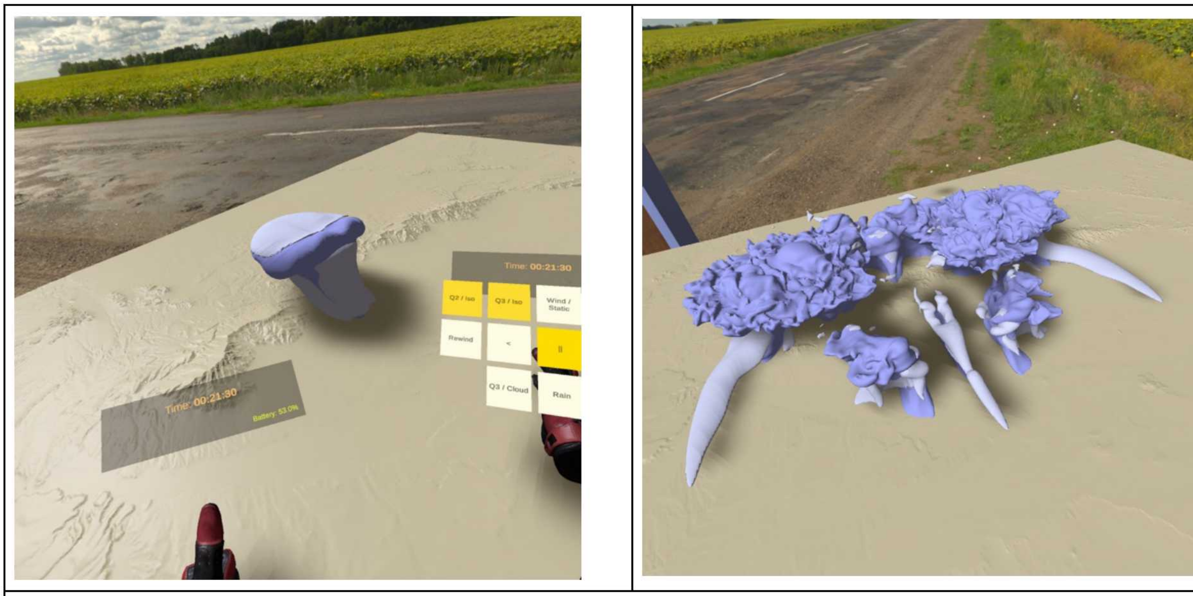


Figure 8. Supercell Thunderstorm in Virtual Reality. The topography on the ground covers a hundred-mile span of the Rio Grande Valley in New Mexico from Santa Fe to Belen.

The COVID-19 Pandemic caused the Idea Lab to be shut down early in the development of VR content for E3SM. The pandemic has made it difficult to demonstrate VR development done by Sandia representatives. While the Idea Lab was open, there were frequent demonstrations of the Super Cell Thunderstorm. VR work continued in homes, but demonstrations were not possible. The Idea Lab has been redesigned for a safer reopening. Equipment and techniques have been added to the Idea Lab procedures to disinfect headsets and other equipment. For example, one single person will be in the Idea Lab room at one time.

New ways to utilize VR and AR will need to be developed. The CARVR Idea Lab has already been redesigned to be useful in a Pandemic world. A cabinet was purchased that with the aid of an alcohol wipe will disinfect an AR or VR headset in a few minutes. Only one person at a time will be in the Idea Lab room. Work was done on creating a generic room that could be used to collaborative on PowerPoint presentations and content for E3SM data. A Vive Pro Eye headset was used to view an Earth-textured object with E3SM simulation added to the sphere. There were also experiments done with mobile AR and with the Hololens 2 headset. This work sets a foundation for future E3SM work. VR is designed to allow people to collaborate over distances. Besides the VR user in the Idea Lab, other users will be in remote locations. Some of these locations will be inside the same building as the Idea Lab and others can be in other Sandia locations or from people homes. Figure 9 is an example of what can be done with mobile AR in the future.

Using AR for E3SM data will require experimentation. There are issues with the complexity of data that can only be resolved through experimentation. It is possible to have remote AR users interact with other AR users and VR users. This is an area that needs more research and development, to determine how valuable it would be for E3SM data.

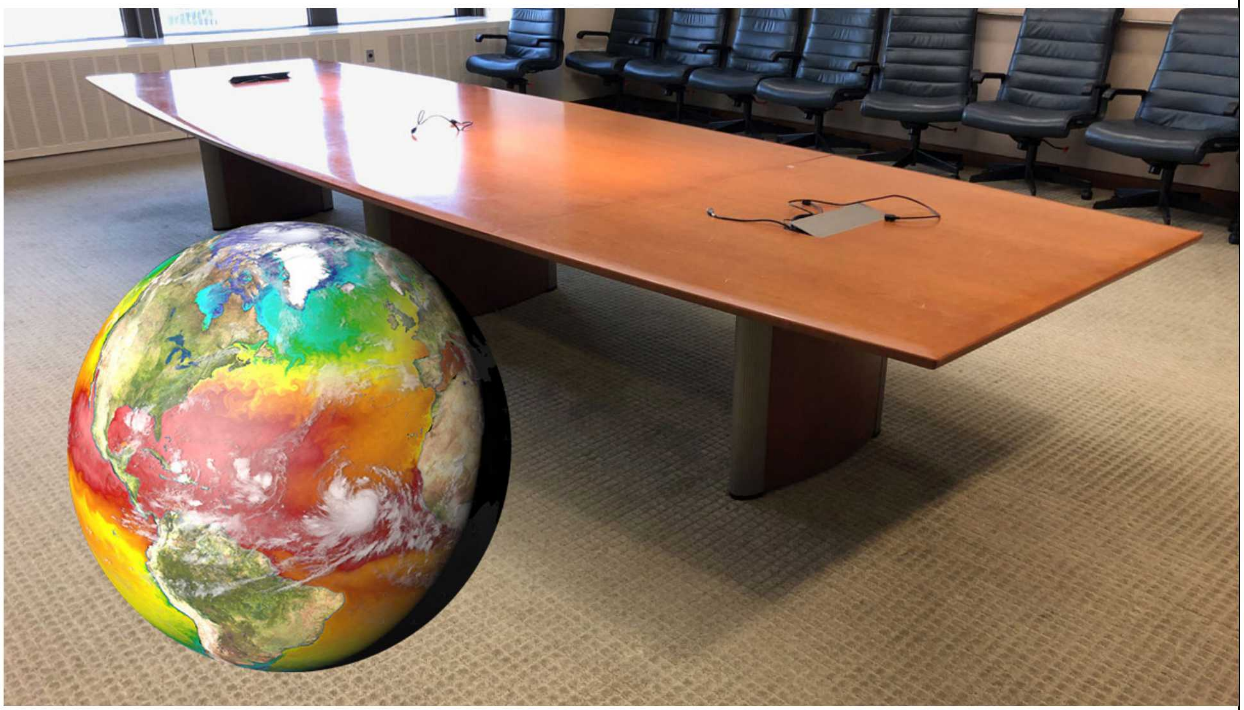
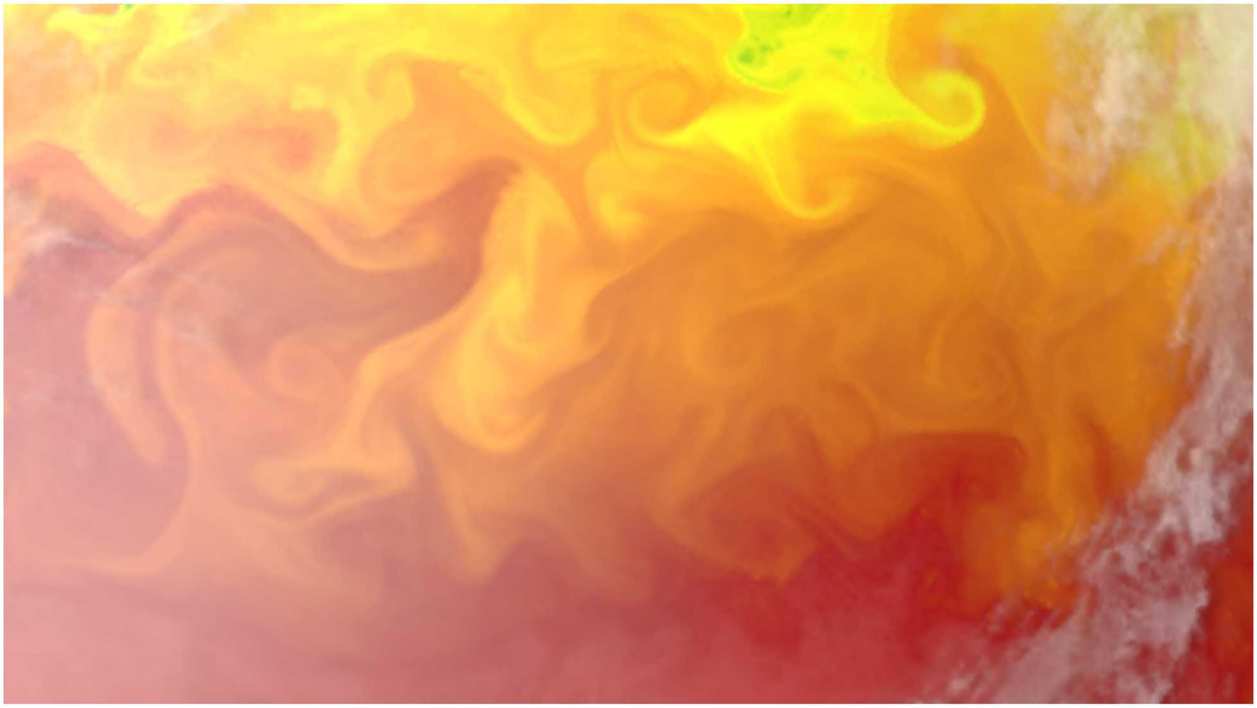


Figure 9. Example of what can be done with mobile virtual experiences in the future as a snapshot of E3SM sea surface temperature data with NASA cloud imagery in a typical conference room.

Section 5 - Realism and Scale Visualizations



Introduction

In this final report for this task includes details regarding the research and development utilizing Houdini Software and Virtual Reality tools for visualization of E3SM Simulation Data and production of a visualization video.

The primary tasks are as follows:

1. Importation and documentation of the importation process of E3SM High Resolution V1 Water cycle simulation data into Houdini software.
2. Creation and documentation of the creation process of an Establishing Shot with a continuous zoom-out from human-scale resolution to global-scale resolution using Google Earth or similar GIS data.
3. Merge Establishing Shot video with E3SM data and document merging process.
4. Create audio track of E3SM visualization video with documentation of audio track process.
5. Composite E3SM simulation data, Establishing Shot and Audio Track into final video for Unclassified Unlimited Release (UUR) to the E3SM community, including documentation of compositing process.
6. Consult with Sandia representatives on incorporation of E3SM simulation data or Non-hydrostatic Thunderstorm simulation into the (Virtual Reality - VR) Idea Lab.

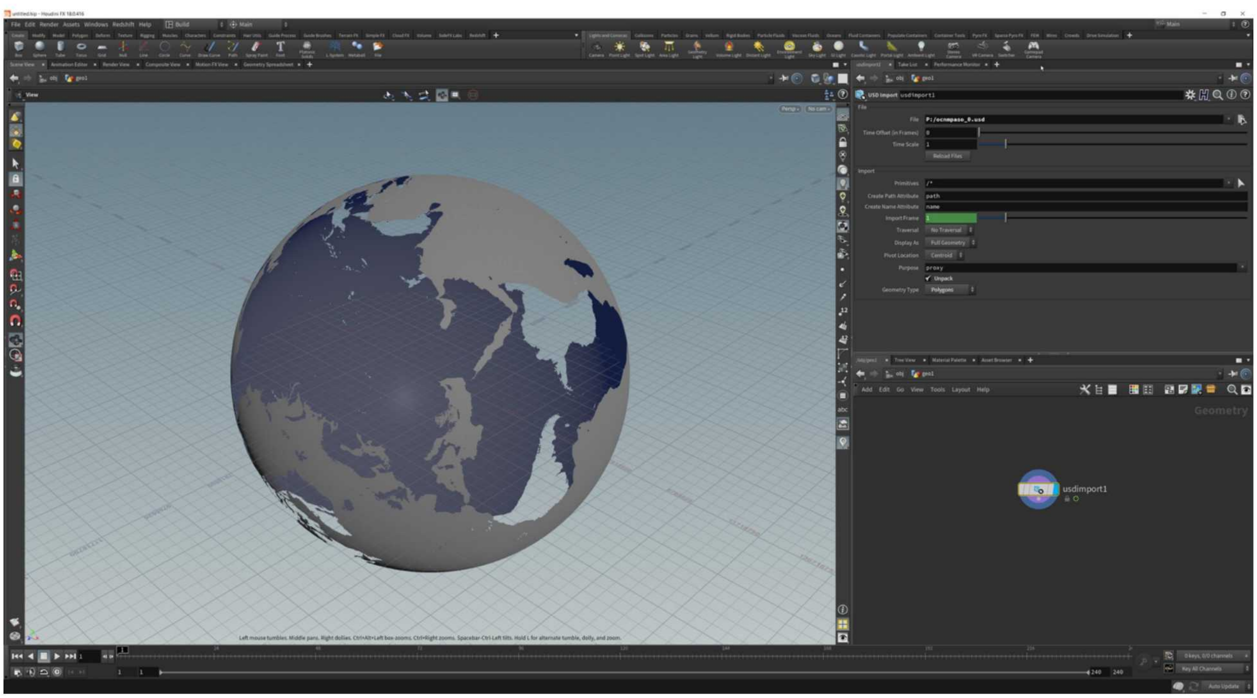
1 - Importation of E3SM Data into Houdini

Task: Importation and documentation of the importation process of E3SM High Resolution V1 Water cycle simulation data into Houdini software

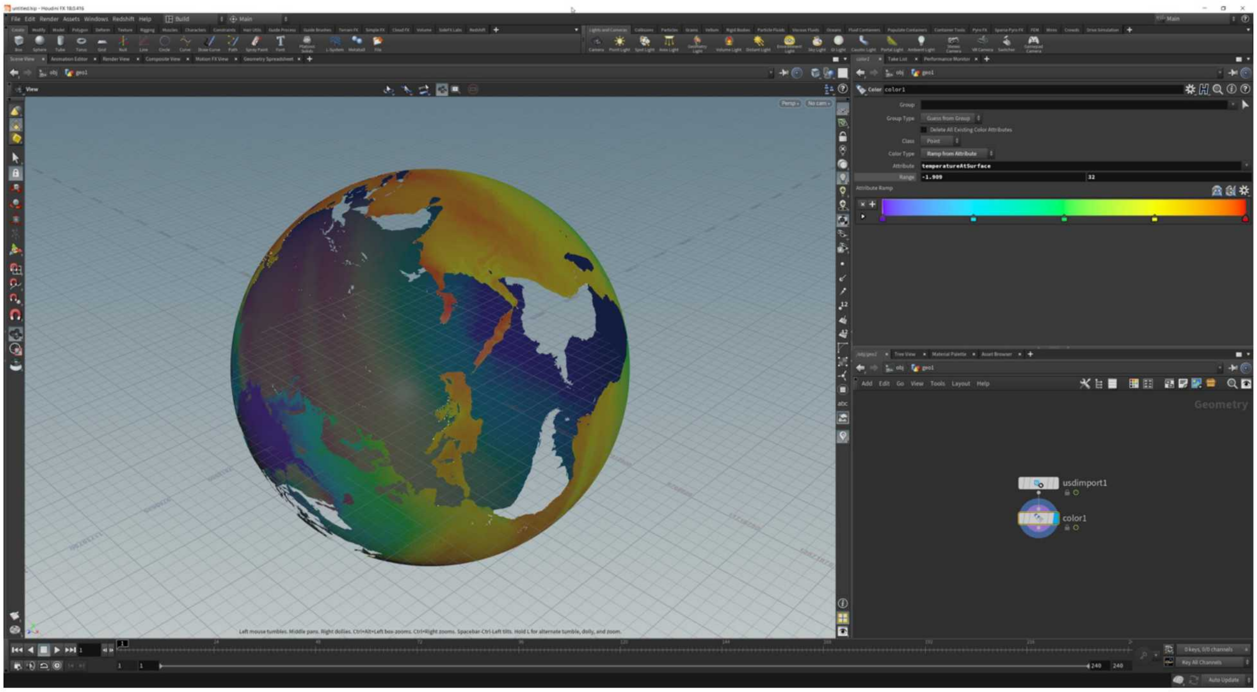
Houdini is a commercial 3D animation application. Houdini is frequently used to create Feature Film visual effects for very complex scenes, like in Marvel's Dr. Strange. Houdini deals with very large datasets, is better than other 3D applications, and was used in prior projects. Houdini's native object format, geo, is different than other object formats. The geo format allows simulation data arrays to be added to primitives /polygons and to points/vertices. Houdini can be used to visualize complex and large simulation datasets using simulation data arrays, like temperature, pressure, velocity and hundreds of others.

A new file format, called USD (Universal Scene Descriptor) can also store simulation data arrays. After the E3SM High Resolution V1 Water cycle simulation data was converted to a USD file format, a single time step of E3SM High Resolution surface temperature data, ocnmpaso_0.usd. I will use that USD file to document the importation process.

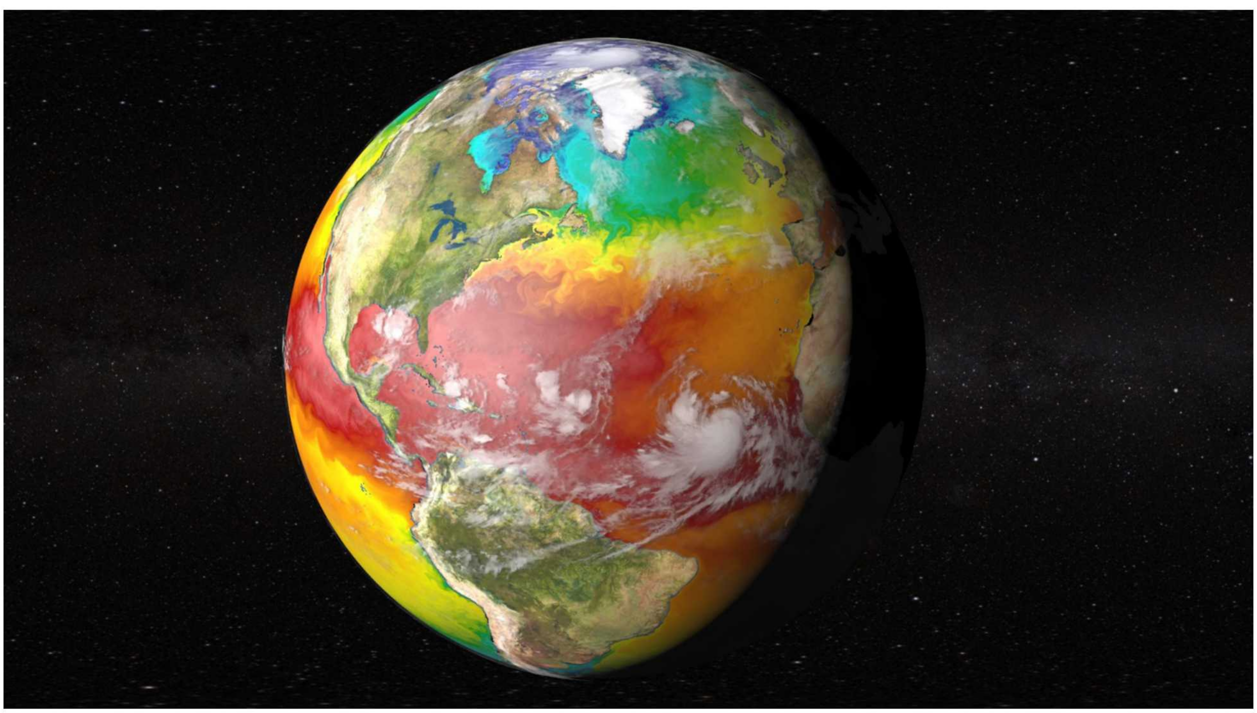
To start an "import USD" node is added to geometry network. A USD file is selected. Unpack is checked and Geometry type is set to Polygons. The USD model is displayed.



Next a color node is added to the USD Import node. The color node allows any of the imported simulation data fields (attributes) to be used to color the imported USD model. An attribute is selected from the attribute drop-down list. The Houdini Geometry Spreadsheet it used to determine the minimum and maximum value for the selected attribute. The ramp can be manually adjusted, or a preset can be selected. In this case, the temperatureAtSurface attribute and the infra-red ramp color preset were selected.



With additional work the colored USD geometry can be added to a textured model of the earth. When volumetric USD data is available, that data could also be added to scene. With modifications to the lighting and layers of E3SM data, Impactful images/animations can be created that appeal to a broader audience.



---

2 - Creation of Establishing Shot

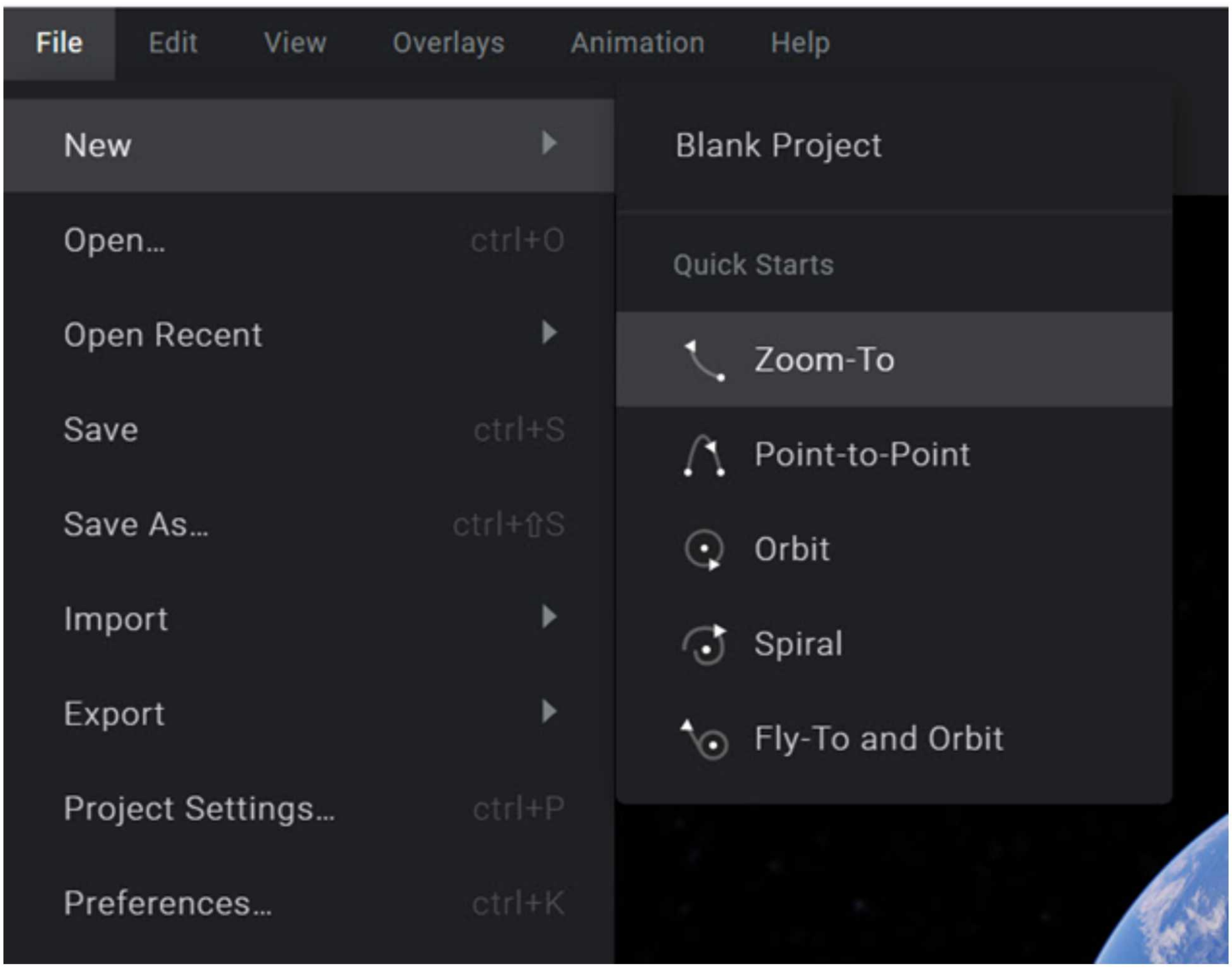
Task: Creation and documentation of the creation process of an Establishing Shot with a continuous zoom-out from human-scale resolution to global-scale resolution using Google Earth or similar GIS data

Google Earth Studio

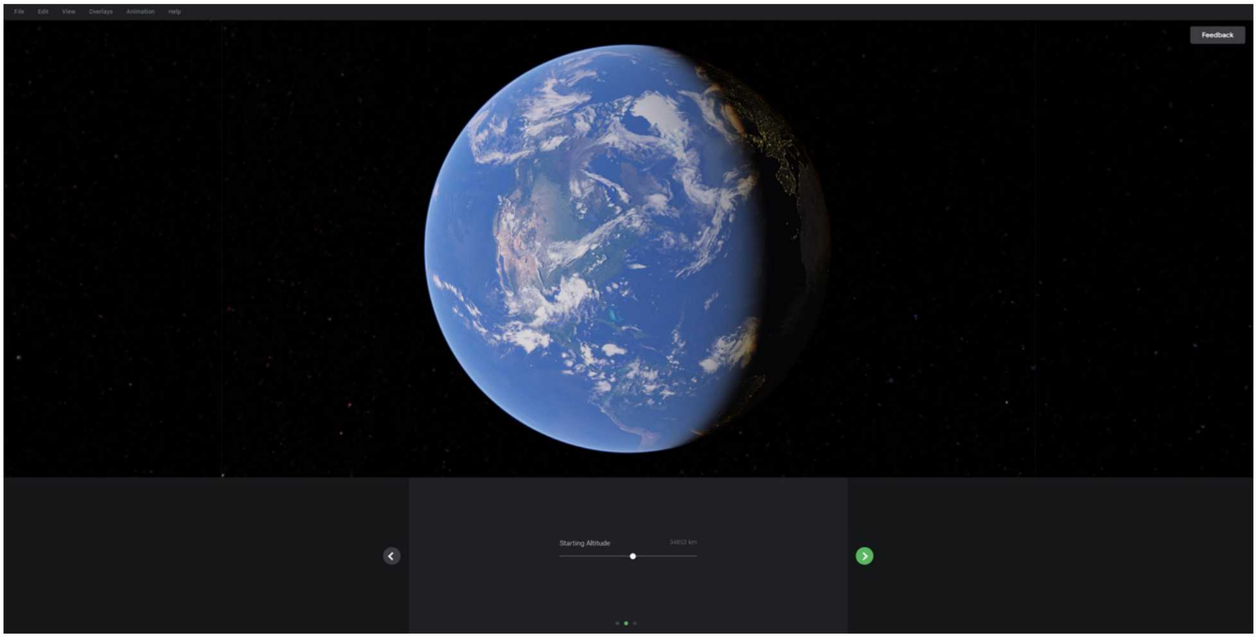
Earth Studio is an animation tool for Google Earth's satellite and 3D imagery. Earth Studio is a web application that makes it very simple to zoom into or zoom out from any place on Earth.

Google Earth Studio is available at the following website: https://www.google.com/earth/studio/

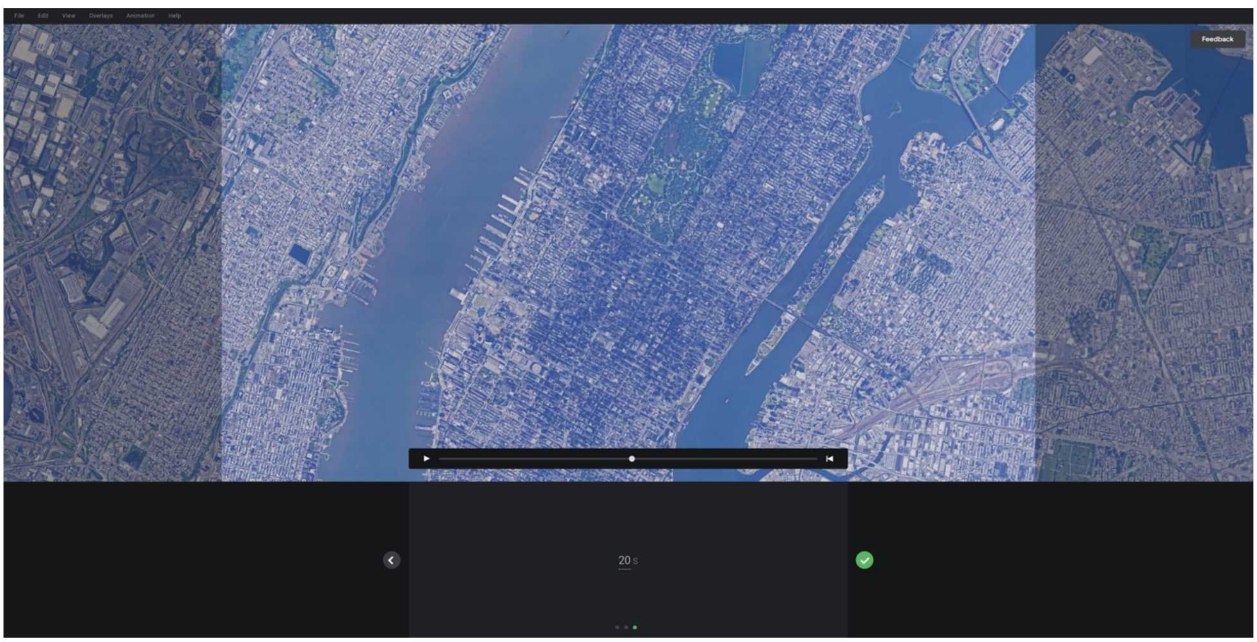
To use Earth Studio, a Google account is required. The app runs in the desktop version of Google chrome. It's free for research, education, film and nonprofit use. After starting Earth Studio, a zoom-in project is created.



The zoom to location is entered, in this case New York City. The location can be Lat/Lon coordinates or the name of a city or location. Next, enter a Starting Altitude. The user can also use the slider to visually adjust the appearance of the earth in the preview window. The altitude of 35,000 km was found to be a good starting point.



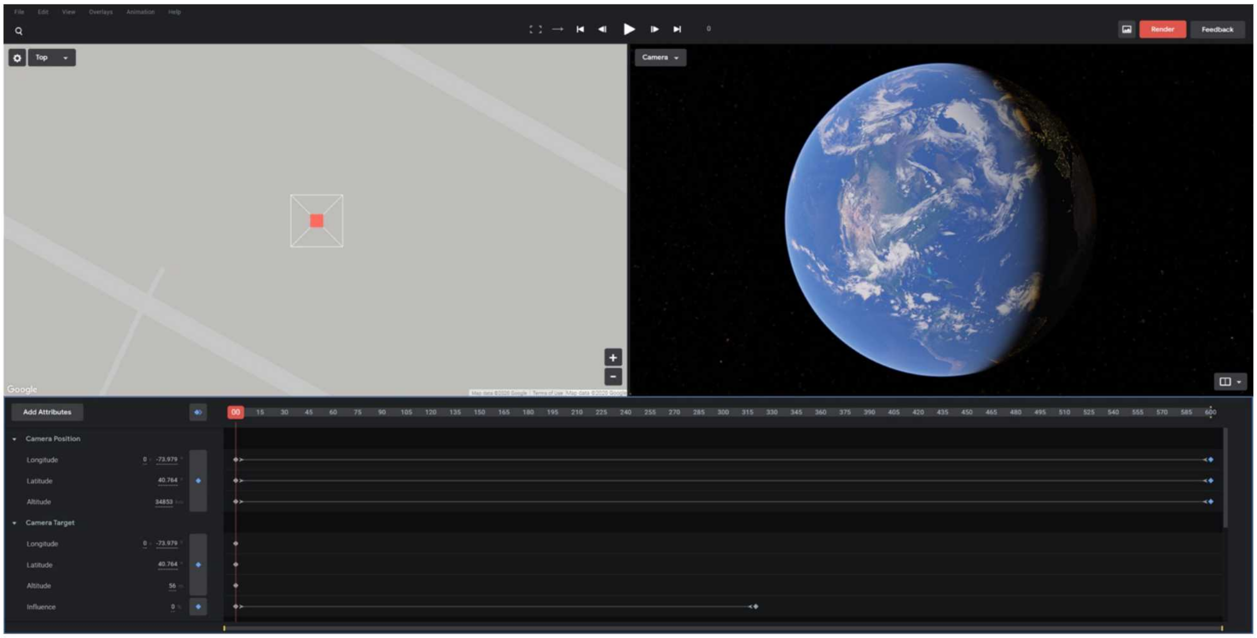
Select the length of the zoom sequence. The default is 5 seconds. To use the zoom as an open sequence for a video, 20-30 seconds is a reasonable length to try.



Slide the Camera Position keyframes to the last frame of the timeline. The Zoom-In project holds the last frame for a few seconds. To allow reversing the sequence, the keyframe must be at the end of the animation.

The first image shows the 3 keyframes in the default location. The second image shows the 3 keyframes moved to the end of the timeline. To move keyframes just click on them and with the left mouse button held down slide them to the right.

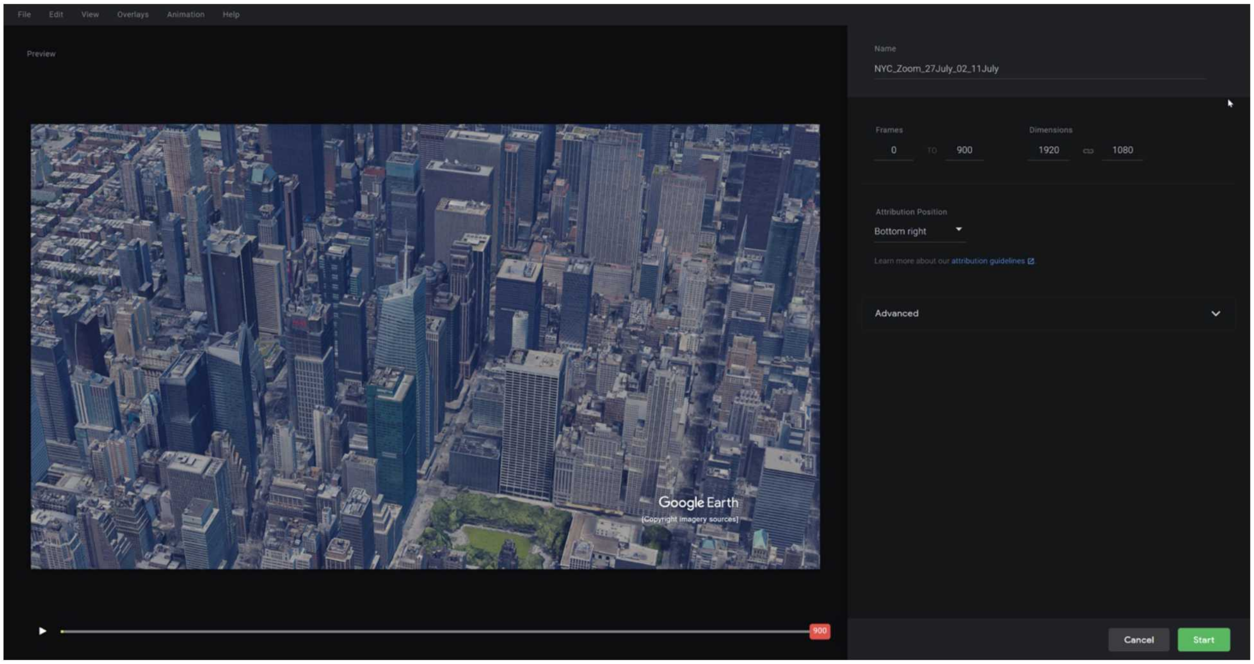
There are other values that the user can change. For example, the time of day will change the Terminator line.



These steps can be used to create a zoom in to any location on Earth. The image sequence can be reversed to create a zoom out. Earth Studio is simple to use and can create a variety of animations. Keyframes can be modified on a variety of properties to animate custom sequences.

To render the zoom sequence, click the red “Render” button on the upper left corner of the screen. The animation will immediately start playing in a loop. If there is something wrong, start over or cancel that screen and try to modify settings in the timeline.

Enter a name for the sequence, and Earth Studio will start rendering the sequence.

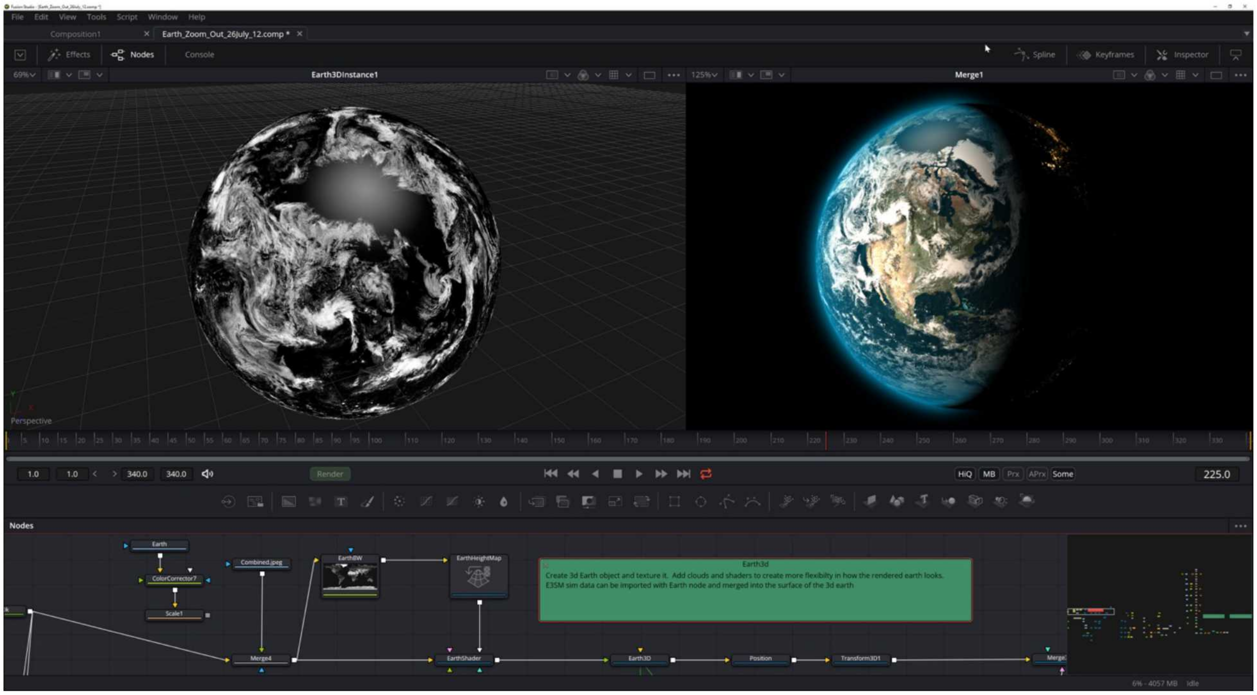


After the sequence is rendered, wait till a zip file is created. The zip file will be saved to the download location for the Chrome browser.

Advanced Establishing Shot

DaVinci Resolve is a video editing application that includes a compositing application, called Fusion. Fusion can be used to create an advanced establishing shot. Preliminary work has been done on creating a Fusion project that will allow E3SM data to be added and rendered.

The Fusion project image below show NASA cloud and surface textures. E3SM atmospheric and surface data can be substituted and rendered.



---

3 - Merge Establishing Shot with E3SM data

Task: Merge Establishing Shot video with E3SM data and document the merging process.

DaVinci Resolve

DaVinci Resolve is a video editing application that can be used to create a video from the establishing shot image sequence, E3SM and other media.

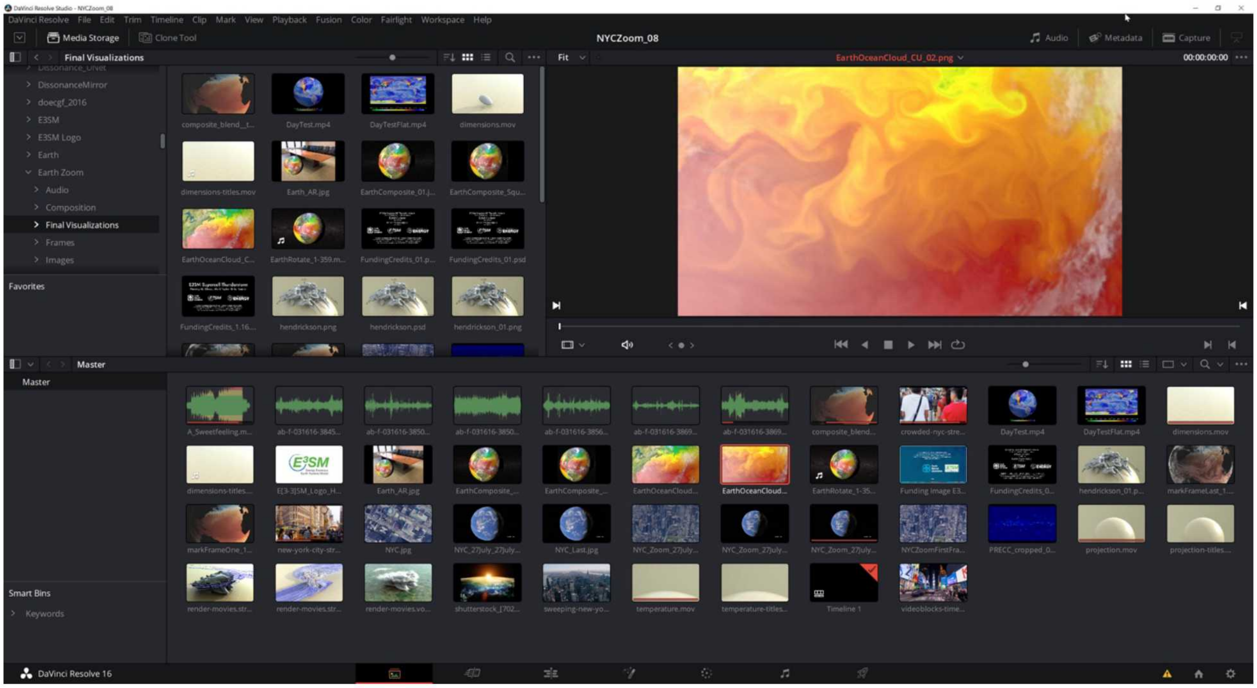
Resolve is available for free at the following website.

https://www.blackmagicdesign.com/products/davinciresolve/

Launch Resolve and a Project Management screen will open. Create a new project and give it a name.

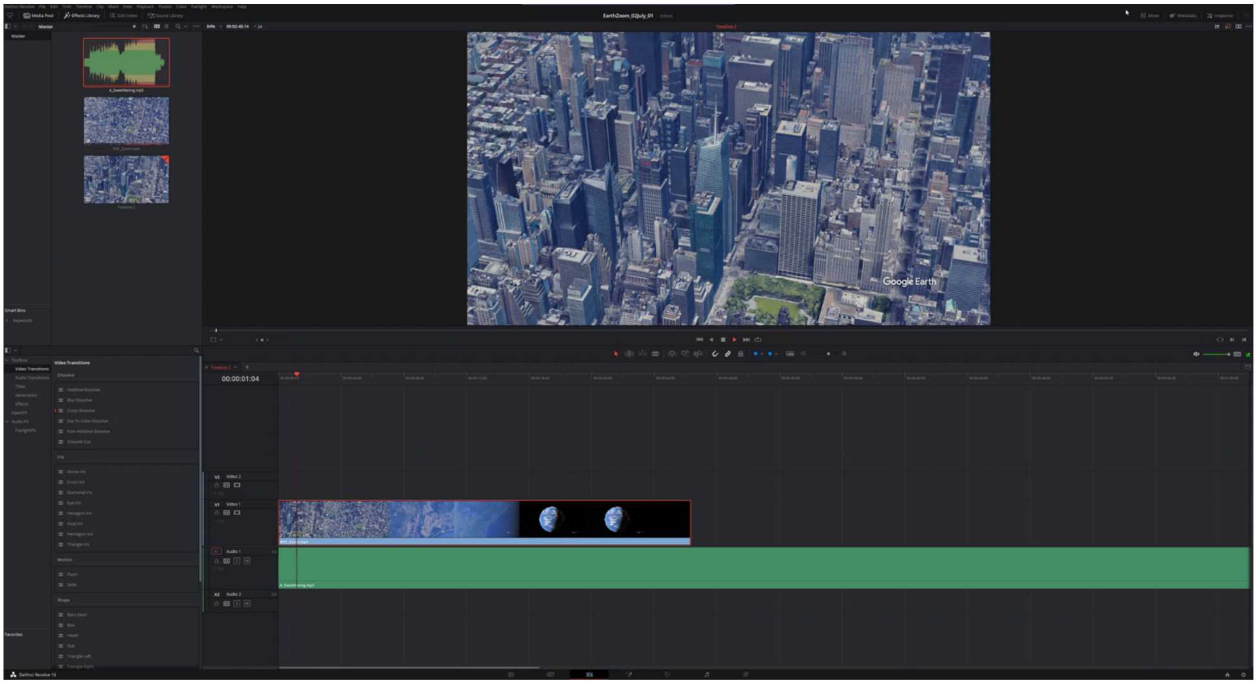
Media

After creating and adding a Project the establishing shot and other media is added to Resolve's media bin. If there is some additional content to add before or after the zoom, it can be dragged and dropped into the media bin. Anytime during the editing of the video, additional content can be dragged and dropped to the media bin. Music, sound effects, stock footage and still images are added to the media bin.



Editing

Merging the establishing shot with E3SM data requires editing. Editing it is part technical and part art. To create a basic edit of the establishing shot with E3SM animations, the shots can be dragged from the Media Bin and dropped on the Resolve timeline, while in EDIT mode. If music has been added to the media bin, that can be dragged and dropped on the timeline. Image below shows a zoom shot of New York City, with music dragged and dropped from the Media bin. Right click on the zoom image sequence and selecting “Change Clip Speed”. When the “Reverse Speed” is checked the clip will become a zoom out instead of a zoom in. This makes is easy to create a zoom in with Earth Studio and use the footage as a zoom in or a zoom out.



Any content that has been added to the media bin can be added to the timeline. Drag and drop E3SM animations and stills to the timeline. Placing them after the zoom shot will add them to the rendered video. To limit the length of the video and to have the music end when the last E3SM animation ends, click on end of the music clip and drag it to the left, until it matches the end of the video clips.

Rendering a final video is done in the Delivery context. Click the Delivery icon on the lower right side of the Resolve screen. Resolve's deliver screen makes it easy to render a video. Select one of the presets on the top left side of the screen. Use the browse button to select a location to save the rendered video to. Then enter a name for the video. Click the add to render queue and your project will be shown on the upper right side of the screen. Click Render start render.

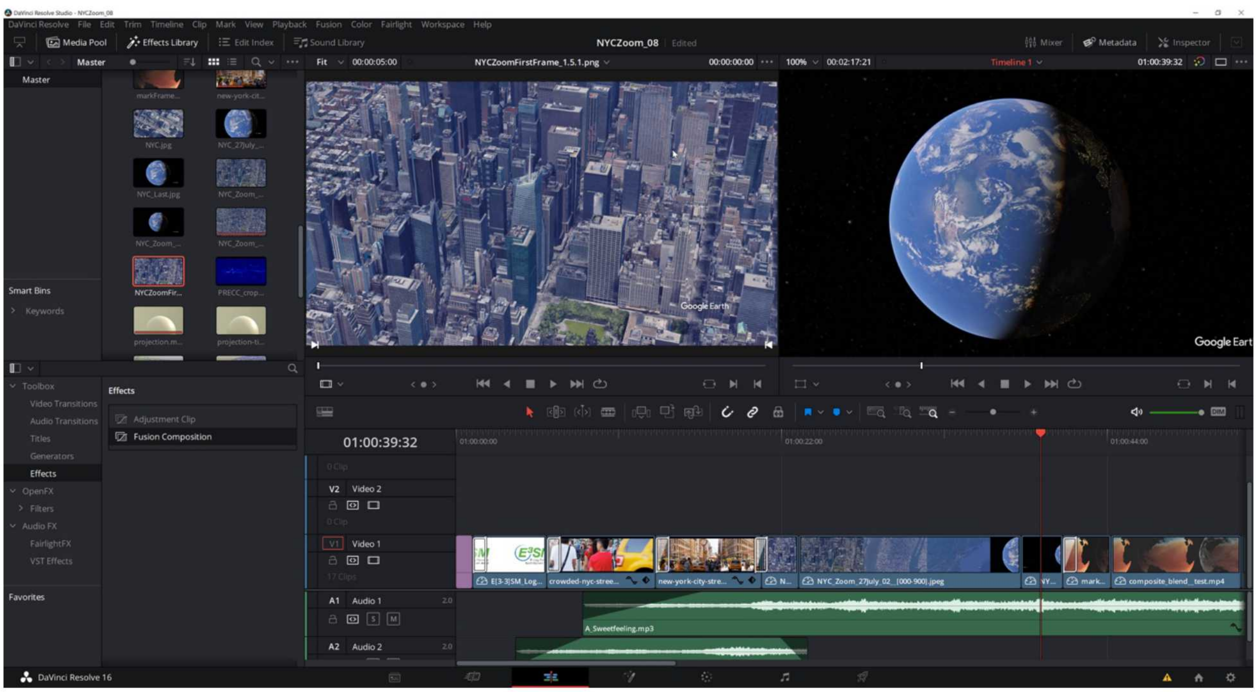
With some editing experience fades, titles, stock footage, narration and sound effects can be added to the video to enhance the overall appear to a broad audience.

---

4 - Create Audio track

Task: Create audio track of E3SM visualization video with documentation of audio track process

Resolve can add as many audio tracks as needed to an editing timeline. Videos can have tracks for music, narration, sound effects and as many more as needed. To add audio to a video, drag and drop audio content from the media bin to the timeline. Designing an effective audio track is part technical and part creative. It can take years of experience to create complex audio tracks. To create an audio track for an establishing shot with E3SM content a music track can be added. Music can be dragged and dropped to the Media bin, then dragged and dropped to the timeline. There are lots of sources for music that can be used in a video. For example, Camtasia has a media bundle that has royalty free music, sound effects and ambient noise samples. The image below shows a video with music and ambient sound added to two audio tracks. The audio levels have been adjusted to match the video clips. The ambient sound has been added to supplement the video clips of people in New York City. Audio clips are move along the timeline to fit the editing of the video. The music fades in as the zoom out starts and the shots of people on the street fades out. The sound of people on the street fades in as the E3SM logo cross fades to the people walking on the street. The E3SM clips are adjusted to change when the music changes. The duration of the first and last frames of the clips can be adjusted to help match the music. Cross fades timed to the music makes the transitions less jarring. In some cases, a fade to black and a fade from black is used instead of a cross fade, to match the transition to the music and to create a break from the continuous fades or cross fades. These are advanced techniques that take experience and experimentation to use effective.



---

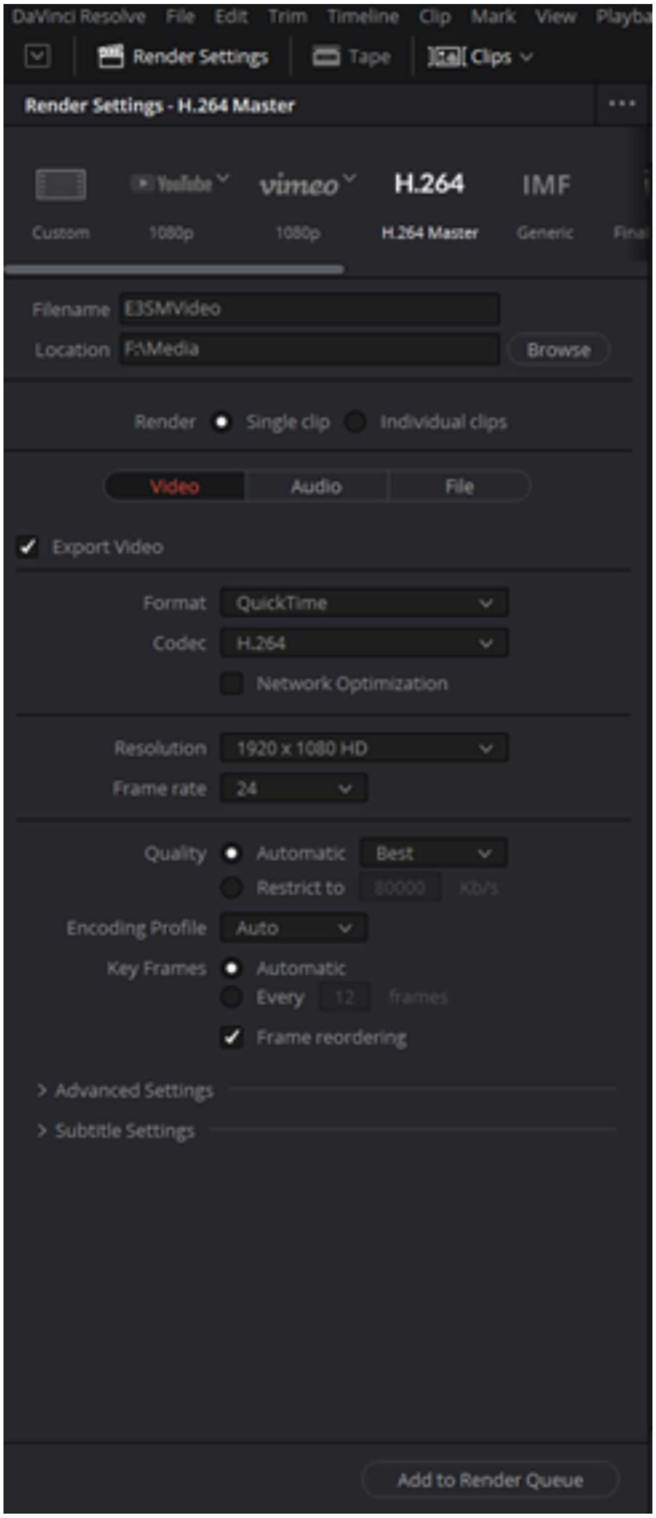
5 - Create UUR Video

Task: Composite E3SM simulation data, Establishing Shot and Audio Track into final video for Unclassified Unlimited Release (UUR) to the E3SM community, including documentation of compositing process

Resolve is used to merge the E3SM simulation, Establishing Shot and the Audio track. This was covered in other tasks. The final video must be rendered and sent through Sandia's R&A process.

Rendering a final video is done in Resolves Delivery context. Click the Delivery icon on the lower right side of the Resolve screen. Resolve's deliver screen makes it easy to render a video. Select one of the presets on the top left side of the screen. Figure 12 shows the H.264 preset selected.   Use the browse button to select a location to save the rendered video to. Then enter a name for the video. Click the add to render queue at the bottom of the Render Settings window. The project is added to the render queue. Changes can be made, or different presets can be selected. Add as many render settings to the render queue as needed. The upper right side of the screen will show all the different versions that are ready to be rendered. Select one or all the versions in the render queue and click Render.

The final video can then be submitted to R&A for classification as UUR.



Conclusion for Realism and Scale

There were issues with translating E3SM data formats. We need to develop techniques that work on multiple platforms. There is hope that Nvidia's Omniverse application will help with translating all type of E3SM data, including volumetric data into USD data files. This is important to make impactful visualizations, including augmented reality and virtual reality, more reproducible by a broader audience. More applications are adopting this standard.

Section 6 - Summary and Next Steps

Throughout the project, we interacted with and received feedback from E3SM Scientists including Mark Taylor, Noel Keen, Mat Maltrud, and Kyle Pressel. Their insight was invaluable and helped shape many of the deliverables. With future work, we plan to strengthen this collaboration to gather their feedback as well as input from the larger E3SM development group including peers, external groups, and sponsors. Our team was new to the E3SM social and technical community, and in this first year, we laid foundations that will lead to rapid progress in future work. Given changes to the project's goals related to the pandemic of developing virtual reality experiences, future work would include safe ways for users to experience virtual and augmented reality. We would continue development of visualizations for the Regional and Global modeling communities such as the E3SM and LASSO projects, and we would continue our work with animations and films expressing realism, scale, and inspiration.