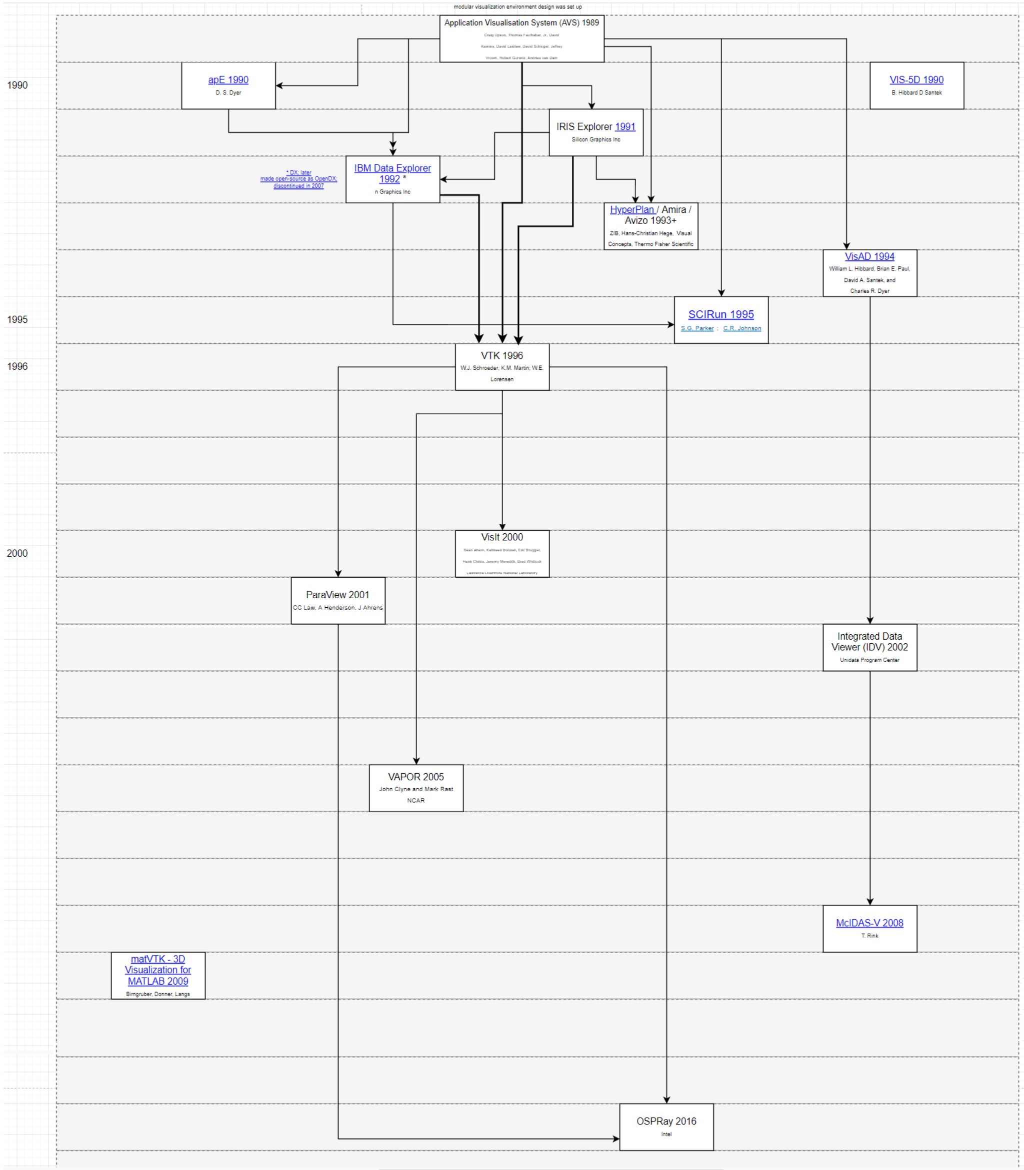
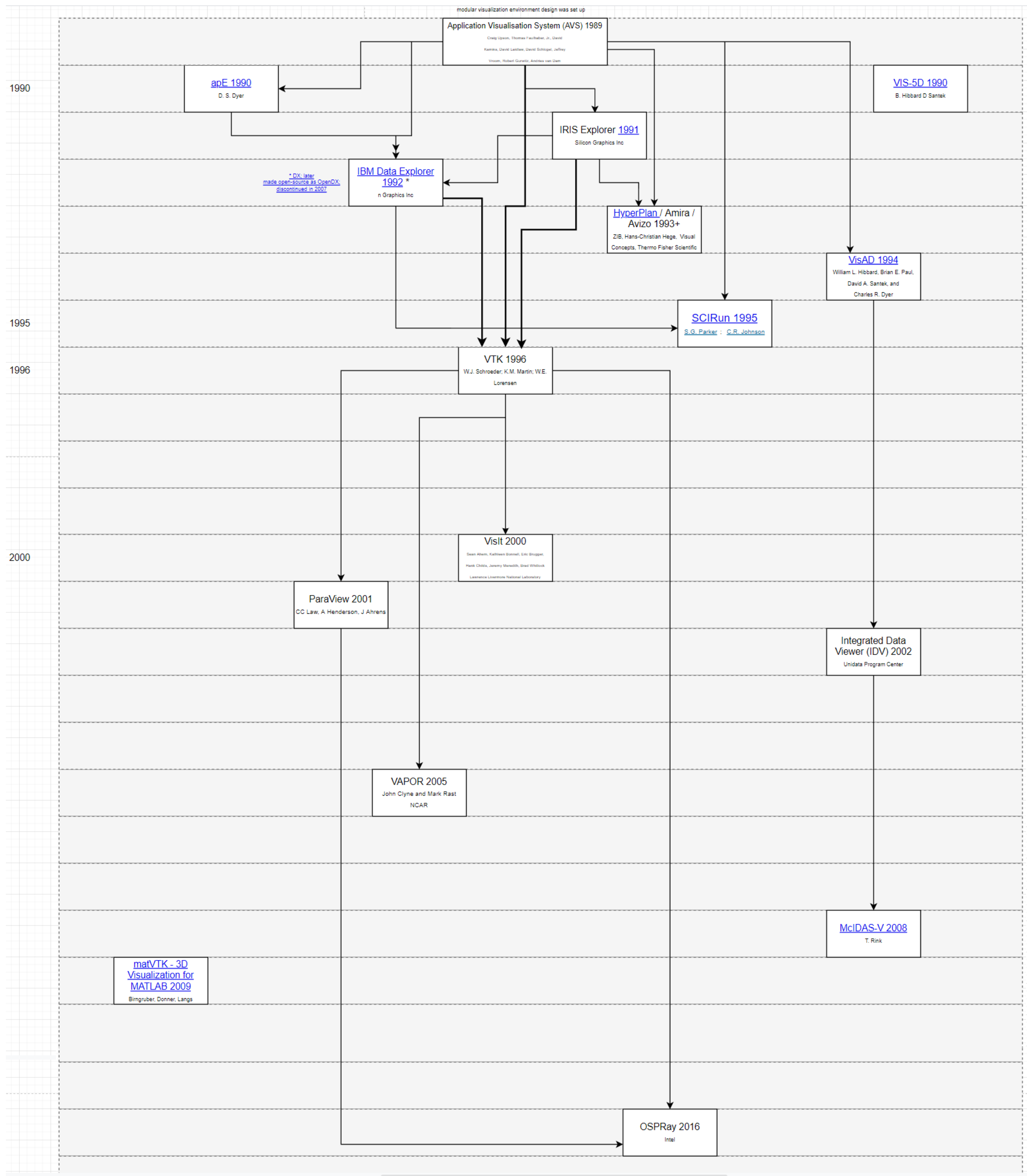
Table 1	ParaView* *built on top of VTK [3]	VTK [4]	Vapor [6] [7]	Intel OSPRay* [11] *library	Blender
+ Pros	*inherits (2), (3), (4) from underlying VTK + Has built-in NetCDF reader + VR Support [19] + Still/Interactive rendering available + “ParaView features are implemented in libraries, it is possible to completely replace the ParaView GUI with your own custom application” [20] this is reminiscent of the AVS modular approach, therefore, there are lots of plug-ins available and developed by the community. + Easy Remote Server-Client setup + Multiple ways of parallelisation (either as	+ IEEE VIS2021 Test of Time Award: many future visualisation toolkits were influenced by VTK, or simply build on top of it. + Supports structured and unstructured data (1) + Linux, Mac OS, Windows support (2) + Scripting support, via TCl and Python [4] (3) + Active ongoing Parallel development (VTK-m)	+ Good performance – OpenGL (GLSL) implementation [1] ➔ macOS, Linux and Windows support + Support of regular and curvilinear grids (efficiently, as Vapor uses separate algorithms) [1] + Support of unstructured models (v 3.1.0) and unstructured model rendering (v3.3.0) [2] + Support for missing values [1] – does not include missing data values when rendering + Blending considers depth information [1] + Interactivity maintained by downgrading rendering during scene navigation [1]	+ Includes a path tracer capable of interactively rendering photorealistic global illumination [12] + Directly integrated into ParaView 5.x. [12] + Focused on HPC visualization rendering, therefore supports both CPU and GPU rendering [11] + “... implementation is specifically designed for CPU based platforms such as Intel Xeon and Xeon Phi based workstations and HPC resources” [11], which are available on ARC3 [13] and ARC4 [14] + Integrated Shading Models [11] + Support for large data (tested on 2048^3)	+ Includes a ray-tracing package (called Cycles) by default + Includes Python scripting +Has a capable realistic rendering engine

	MPI, or batch/array jobs with pvbatch) + Significant availability of data analysis toolkits (TTK, VortexFinder) and parallel algorithms (VTK-m)		+ Support of large datasets – 2048^3 was manageable as far back as 2007 [6]		
– Cons	- Realistic rendering (with cast-shadows) is limited to the built-in OSPRay library - Complicated Build process (a lot of options, need to research and include them before building and set-up)	– Is a toolkit, therefore, would require building the user interface for it [3] – Not easy to set up [3]	– No support of scripting [8] [9] as of 2022 Release 3.6.0 [10]	– Not faster than OpenGL GPU solutions, but competitive and similar results when using GPU [11] (note that this library focuses on CPU solutions)	– Main focus on film-production or video game model creation – As software not tailored for scientific visualisation, not scalable for large data visualisations
Notes	ParaView is used unanimously in the scientific community and has active ongoing development.		While initially started out as a collection of software and hardware technologies [7], and used the VTK ray caster, now includes custom collection of C++ class objects that implement various 2D and 3D visualization Algorithms [1]	This is a library, not a separate application or toolkit. [11] Main novelty is “a fast CPU-based alternative to Mesa and OpenGL visualization” [11]	

Table 2	McIDAS-V	VisIt *built upon VTK	IDL	MATLAB
+ Pros	- Known in meteorology community	+ Contains a GUI + Supports rectilinear, curvilinear, unstructured and structured data + Has Python Scripting interface + Extensible via plug-ins + Has good HPC support [17] + Wide support of scientific data formats [18]	+ Fundamentally scriptable + Wide support for meteorological data formats (including HDF, CDF, NCDF) + Has direct and indirect volume visualisation	+ Fundamentally scriptable
– Cons	– Primary focus is on environmental satellite data visualisation, rather than simulation data, therefore lacks more advanced 3D rendering	- No globally-uniform lighting computation (using Phong Shading instead) - More complex client/server remote usage setup[22]	- Steeper learning curve - Lack of realistic visualisation capabilities	– If Volume Rendering is desired, it needs to be extended with matVTK to get 3D Visualization which is a MATLAB-VTK interface
Notes	Sequence of development: 1. VisAD *implemented as generalised (now discontinued) Vis-5d 2. Integrated Data Viewer (IDV) - Unidata* *built upon the VisAD library			

Timeline 1


modular visualization environment design was set up
Application Visualisation System (AVS) 1989
Craig Upson, Thomas Faulhaber, Jr., David
Kamins, David Laidlaw, David Schlegel, Jeffrey
Vroom, Robert Gurwitz, Andries van Dam
1990
apE 1990
D. S. Dyer
VIS-5D 1990
B. Hibbard D Santek
IRIS Explorer 1991
Silicon Graphics Inc
* DX: later
made open-source as OpenDX;
discontinued in 2007
IBM Data Explorer
1992 *
n Graphics Inc
HyperPlan / Amira /
Avizo 1993+
ZIB, Hans-Christian Hege, Visual
Concepts, Thermo Fisher Scientific
VisAD 1994
William L. Hibbard, Brian E. Paul,
David A. Santek, and
Charles R. Dyer
1995
SCIRun 1995
S.G. Parker ; C.R. Johnson
1996
VTK 1996
W.J. Schroeder; K.M. Martin; W.E.
Lorensen
2000
VisIt 2000
Sean Ahern, Kathleen Bonnell, Eric Brugger,
Hank Childs, Jeremy Meredith, Brad Whitlock
Lawrence Livermore National Laboratory
ParaView 2001
CC Law, A Henderson, J Ahrens
Integrated Data
Viewer (IDV) 2002
Unidata Program Center
VAPOR 2005
John Clyne and Mark Rast
NCAR
McIDAS-V 2008
T. Rink
matVTK - 3D
Visualization for
MATLAB 2009
Bimgruber, Donner, Langs
OSPRay 2016
Intel



modular visualization environment design was set up
Application Visualisation System (AVS) 1989
Craig Upson, Thomas Faulhaber, Jr., David
Kamins, David Laidlaw, David Schlegel, Jeffrey
Vroom, Robert Gurwitz, Andries van Dam
1990
apE 1990
D. S. Dyer
VIS-5D 1990
B. Hibbard D Santek
IRIS Explorer 1991
Silicon Graphics Inc
* DX: later
made open-source as OpenDX;
discontinued in 2007
IBM Data Explorer
1992 *
n Graphics Inc
HyperPlan / Amira /
Avizo 1993+
ZIB, Hans-Christian Hege, Visual
Concepts, Thermo Fisher Scientific
VisAD 1994
William L. Hibbard, Brian E. Paul,
David A. Santek, and
Charles R. Dyer
1995
SCIRun 1995
S.G. Parker ; C.R. Johnson
1996
VTK 1996
W.J. Schroeder, K.M. Martin, W.E.
Lorensen
2000
VisIt 2000
Sean Ahern, Kathleen Bonnell, Eric Brugger,
Hank Childs, Jeremy Meredith, Brad Whitlock
Lawrence Livermore National Laboratory
ParaView 2001
CC Law, A Henderson, J Ahrens
Integrated Data
Viewer (IDV) 2002
Unidata Program Center
VAPOR 2005
John Clyne and Mark Rast
NCAR
McIDAS-V 2008
T. Rink
matVTK - 3D
Visualization for
MATLAB 2009
Bimgruber, Donner, Langs
OSPRay 2016
Intel

Based on Tables 1, 2 and the Timeline 1, it is evident that a set of two distinct research cycles appear. First one starts at the initial set-up of the modular visualization environment design with the introduction of Application Visualisation System (AVS) in 1989. The other one – after the conception of VTK. The modern choices for scientific visualisation either stem from adding to / building from VTK, or acknowledging VTK, yet implementing custom tools. One such framework is the NCAR VAPOR package [1] [2], which initially started as a collection of software and hardware technologies, which included VTK in the very first prototype [7], but diverged from it to [21].

[1] @article{Li2019VAPORAV,

```
  title={VAPOR: A Visualization Package Tailored to Analyze Simulation Data in Earth System Science},
  author={Shaomeng Li and Stanislaw Jaroszynski and Scott Pearse and Leigh Orf and John P. Clyne},
  journal={Atmosphere},
  year={2019}
}
```

[2] https://vapor.readthedocs.io/en/readthedocs/downloads/officialReleases.html

[3] @article{ahrensapplication,

```
  title={An Application Architecture for Large Data Visualization: A Case Study},

  author={Ahrens, James}
```

}

[4] @INPROCEEDINGS{567752,

author={Schroeder, W.J. and Martin, K.M. and Lorensen, W.E.},

booktitle={Proceedings of Seventh Annual IEEE Visualization '96},

title={The design and implementation of an object-oriented toolkit for 3D graphics and visualization},

year={1996},

volume={},

number={},

pages={93-100},

doi={10.1109/VISUAL.1996.567752}}

[5] https://gitlab.kitware.com/vtk/vtk

[6] '@article{clyne2007interactive,

title={Interactive desktop analysis of high resolution simulations: application to turbulent plume dynamics and current sheet formation},

author={Clyne, John and Mininni, Pablo and Norton, Alan and Rast, Mark},

journal={New Journal of Physics},

volume={9},

number={8},

pages={301},

year={2007},

publisher={IOP Publishing}

}

[7] '@inproceedings{clyne2005prototype,

  title={A prototype discovery environment for analyzing and visualizing terascale turbulent fluid flow simulations},

  author={Clyne, John and Rast, Mark},

  booktitle={Visualization and Data Analysis 2005},

  volume={5669},

  pages={284--294},

  year={2005},

  organization={SPIE}

}

[8] https://github.com/NCAR/VAPOR/issues/1894

[9] https://github.com/NCAR/VAPOR/issues/1908

[10] https://github.com/NCAR/VAPOR/releases/tag/3.6.0

[11] @ARTICLE{7539599,

  author={Wald, I and Johnson, GP and Amstutz, J and Brownlee, C and Knoll, A and Jeffers, J and Günther, J and Navratil, P},

  journal={IEEE Transactions on Visualization and Computer Graphics},

  title={OSPRay - A CPU Ray Tracing Framework for Scientific Visualization},

  year={2017},

  volume={23},

  number={1},

  pages={931-940},

doi={10.1109/TVCG.2016.2599041}}

[12] https://www.ospray.org/

[13] https://arcdocs.leeds.ac.uk/systems/arc3.html

[14] https://arcdocs.leeds.ac.uk/systems/arc4.html

[15] @techreport{moreland2008large,
title={Large Scale Visualization with ParaView (slides).},
author={Moreland, Kenneth D and Greenfield, John Andrew and Scott, W Alan and Ayachit, Utkarsh and Geveci, Berk and DeMarle, David},
year={2008},
institution={Sandia National Lab.(SNL-NM), Albuquerque, NM (United States)}
}

[16] @ARTICLE{299413,

author={Hibbard, W.L. and Paul, B.E. and Santek, D.A. and Dyer, C.R. and Battaiola, A.L. and Voidrot-Martinez, M.-F.},

journal={Computer},

title={Interactive visualization of Earth and space science computations},

year={1994},

volume={27},

number={7},

pages={65-72},

doi={10.1109/2.299413}}

[17] @article{osti_15004017,

title = {VisIt: a component based parallel visualization package},

author = {Ahern, S and Bonnell, K and Brugger, E and Childs, H and Meredith, J and Whitlock, B},

abstractNote = {We are currently developing a component based, parallel visualization and graphical analysis tool for visualizing and analyzing data on two- and three-dimensional (20, 30) meshes. The tool consists of three primary components: a graphical user interface (GUI), a viewer, and a parallel compute engine. The components are designed to be operated in a distributed fashion with the GUI and viewer typically running on a high performance visualization server and the compute engine running on a large parallel platform. The viewer and compute engine are both based on the Visualization Toolkit (VTK), an open source object oriented data manipulation and visualization library. The compute engine will make use of parallel extensions to VTK, based on MPI, developed by Los Alamos National Laboratory in collaboration with the originators of P K . The compute engine will make use of meta-data so that it only operates on the portions of the data necessary to generate the image. The meta-data can either be created as the post-processing data is generated or as a pre-processing step to using VisIt. VisIt will be integrated with the VIEWS' Tera-Scale Browser, which will provide a high performance visual data browsing capability based on multi-resolution techniques.},

doi = {},

url = {https://www.osti.gov/biblio/15004017}, journal = {},

number = ,

volume = ,

place = {United States},

year = {2000},

month = {12}

}

[18] http://www.visitusers.org/index.php?title=Partial_list_of_file_formats_and_their_details_VisIt_supports

[19] https://www.kitware.com/navigation-basics-in-virtual-reality-with-paraview/

[20] '@misc{moreland2007large,

title={Large Scale Visualization with ParaView 3},

author={Moreland, Kenneth and Greenfield, John},

year={2007}

}

[21] http://cscads.rice.edu/alan-norton-cscads-2009.pdf

[22] https://visit-sphinx-github-user-manual.readthedocs.io/en/develop/tutorials/RemoteUsage.html