Workshop on Visualisation in Environmental Sciences (EnvirVis) (2018)
R. Bujack and K. Rink and D. Zeckzer and S. Jänicke (Editors)

Web-based 3D Meteo Visualization: 3D Rendering Farms from a New Perspective

M. Koutek[1] and I. van der Neut[1]

[1]KNMI, The Royal Netherlands Meteorological Institute, The Netherlands.

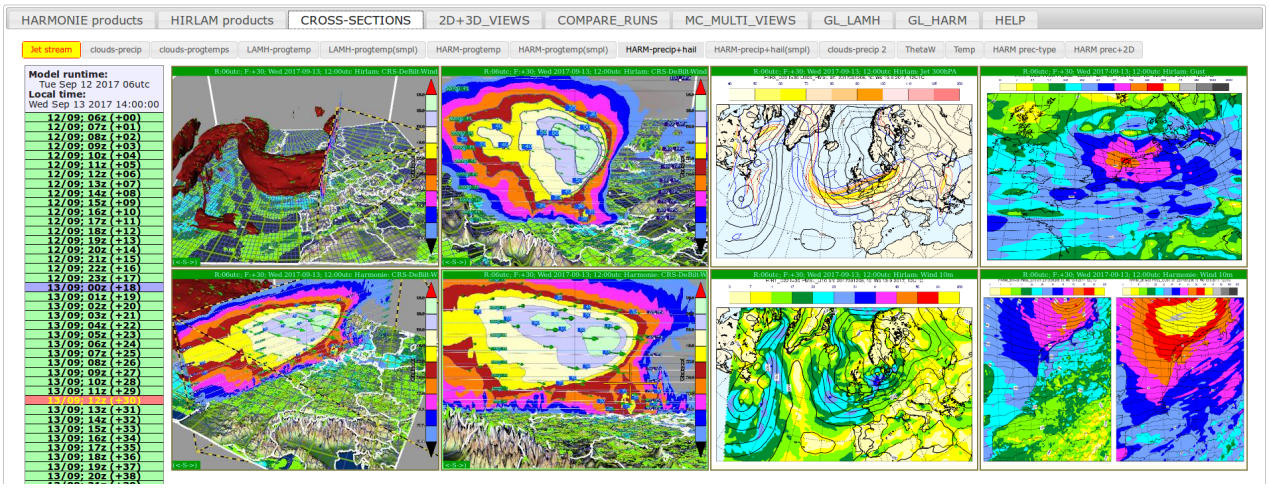

Figure 1: *W3DX Web-Portal: Exploring a wind storm. For two weather models (HIRLAM and HARMONIE) showing simultaneously the upper-air and surface interaction. 3D images of jet-stream with an overview camera and a camera perpendicular to the cross-section (left); 2D images of jet-stream contours at 300hPa with isobaric line on ground, wind and wind-gusts at 10m (right). Images of 3D visualization products for any forecast-time are shown in the web-portal as quickly as images of the conventional 2D meteo-visualizations.*

Abstract

We present an approach to highly responsive and interactive web-based 3D meteorological visualization. We describe important technical aspects of the current proof-of-concept implementation that leverages a 3D rendering farm. The work has been motivated by high demands of operational weather forecasters on ease of use of such a tool, quick system responsiveness and low operational end-to-end latency, instantaneous navigation in forecast-time (max. 5 seconds to navigate over 50 time-steps), and other challenging requirements considering the meteo data volumes and complexity of 3D visualization processing. We had limited computer resources but were allowed to spend any time needed in the back-end processing to maximize the responsiveness of the front-end (web-portal).

Our goal was not to implement a perfect system, but to design a suitable architecture and implement a responsive system filled with 2D and 3D visualization products of operational NWP (numerical weather prediction) model runs for testing by forecasters and discovering good use-cases for 3D in operations. We succeed in bringing conventional 2D and 3D visualizations together in one web-portal. Customized visualizations layouts can be created to depict a certain atmospheric phenomenon. The web-portal provides display of all 3D products (pre-rendered) also as VR (virtual reality) images through the web-browser. The user can instantaneously rotate the camera-view on 3D products and rotate the vertical cross-sections in the web-browser. The system when used on a 3D high-end workstation with VR display provides fully interactive VR data exploration of any given 3D visualization product shown in the 3D preview in the web-portal.

Although from today's perspective (modern browsers, Web-GL and cloud 3D computing/visualization technologies) our approach of rigorously pre-rendering 3D products might seem unnecessary, the contrary is true. The pre-rendered 3D views off-line in back-end (pre-caching) make sure the user will not suffer a poor web-portal responsiveness on-line.

CCS Concepts

•***Computing methodologies*** → ***Scientific visualization; Rendering; Graphics systems and interfaces;*** *Parallel computing methodologies;* •***Human-centered computing*** → ***Visualization;*** *Visualization toolkits;*

DOI: 10.2312/envirvis.20181132

1. Motivation

Traditionally, operational weather forecasting is dominated by the use of automated and interactive visualization systems. Forecasters rely merely on 2D visualization methods, explore different phenomena, look at images, compare output of NWP (numerical weather prediction) model data with observations and measurements, and compare results of different NWP models in order to create an actual mental 3D model of the atmosphere [HC04]. The real atmosphere behavior is very complex and conceptual models simplify the processes. Many of the 2D visualizations show only important parts of the data depicting the given phenomenon without too much visual clutter that could hinder the human observer in effective visual analysis of the images.

The NWP models describe a 3D state of atmosphere. The model simulations produce large time-dependent volumetric datasets which have been studied for several decades by scientific 3D visualization methods, for example VIS5D [HAF*96], VAPOR [CMNR07], or Met3D [RKSW15]. 3D visualization techniques are able to reveal spatial relations in the atmospheric datasets which would not be visible with common 2D visualization methods in horizontal and vertical cross-sections, in data-field on the ground-level, on a certain pressure or a flight level for example. The historically proven solutions from other domains (CAD, architecture, medical visualization) combine 3D and 2D seamlessly. There exist various professional- and academic-grade visualization software for (oceanographic and) atmospheric data which intends to provide exactly that. An example is the open-source Paraview [AGL05] which is based on Visualization Toolkit. VTK is very popular among scientists and has undergone a great metamorphosis on the rendering part with OpenGL [HMCA15].

Showing and exploring 3D visualizations in a Virtual Reality (VR) environment is often overwhelming [HBR*14], [KvdNL*11], [vDLS02]. The latest Computer Graphics and Visualization technologies provide us, as computer scientists, with means to show much more data, more quickly on our displays. Even game engines are being utilized for building VR visualization environments [RBK17]. Despite technological advances, there are limits on how many 3D-shaded structures a human can individually recognize as features and track in time. Showing phenomena in 3D can create occlusions in the scene and eventually hide other important features. Also due to 3D rendering with perspective projection, it becomes hard for the users to locate / deduct a 2D position on a map of a 3D feature hanging somewhere in the air. We have to admit there are some aspects of 3D visualization which are confusing to forecasters. Daily routine use of 3D tooling in operational weather forecasting remains a great challenge. It has been tried in the past two decades with varying (partial) degrees of success [KSH*98], [Hib05], [HSUB89], [MMS00], [MM07].

We research where the forecasters could benefit from 3D visualization and what prevents them from using it. It should be made clear that there is a great difference between an operational forecaster on duty and a forecaster or a meteorologist in a non-operational role. The duty forecasters work in a (time-)constrained environment (Figure 7) and will not use (3D) visualization systems which are complicated in use, differ too much from other systems, have unclear added value for operations, and which take much effort to start or to navigate to another forecast-time [KDvdN16].

In the past ten years there has been enormous growth of new NWP models and model versions. A typical meteorological institute runs multiple different (local area) NWP model simulations per day. Operational forecasters continuously look at NWP data from the own production systems as well as NWP data from neighboring countries and global NWP model data, in Europe provided for example by ECMWF (The European Centre for Medium-Range Weather Forecasts). At KNMI (The Royal Netherlands Meteorological Institute) our operational production includes the HIRLAM NWP model, delivering data 4x per day with forecast up to 48 hours, and the HARMONIE NWP model, delivering data 8x per day with forecast up to 48 hours and with 1 hour time-steps. When new NWP model versions, as happens with the HARMONIE model [BAA*17], are introduced to operational forecasters, appropriate 3D visualization methods can help to understand changes in model physics, convective schemes, or radiation and surface schemes. The HIRLAM model (version 7.2) covers a very large area (Figure 1 top-row) and has a grid resolution of 11 kms and 60 vertical levels. The HARMONIE model (version 36) version covers a smaller area but has a grid resolution of 2.5 kms and 60 vertical levels. The latest HARMONIE versions have more vertical levels. The HARMONIE model has a greatly improved physics and it provides a number of extra atmospheric variables that (old) HIRLAM does not have. The raw NWP model output of these models contains typically 100-200 variables, with some available only at a certain 2D (height or pressure) level or at a surface, and about 10-14 fully 3D variables suitable for 3D visualization. The individual forecast files are delivered in a compressed form (GRIB format) with file sizes varying between 600 MB up to 1GB. This requires use of efficient data handling strategies.

2D meteo-visualizations have been massively automated by pre-rendering thousands of images with conventional meteorological products which are offered to the forecasters in different web-portals. There are currently initiatives that use WMS (Web Mapping Services) and provide 2D layered meteo-visualizations online on-demand through web-portals. For interactive (2D) use is the raw NWP model data converted into NetCDF format, stored on network data servers (i.e. THREDDS), and accessed via APIs such as OpenDAP. For example the KNMI has developed the ADAGUC GeoServices which are being utilized in the GEOWEB visualization platform [PdVT*18]. The forecasters very quickly accommodated web-portal use, which contrasts with the poor adoption of interactive meteo-visualization workstations. They know where to find a certain type of product (no preparation time needed) in the web-portal menu. They need to be able to navigate very quickly through the forecast-time (within 5 seconds scroll over all time-steps), going back-and-forth in time studying the (2D) dynamics of the meteorological features. It appears to be a great challenge to deliver 3D meteo-visualizations so quickly through a web-portal. Especially if we consider multiple concurrent users and several different 3D visualization products or NWP models displayed in the web-portal for each user. This is where the state-of-the-art remote (parallel) visualization systems like Paraview [AGL05] or IDV [Fis17] in the cloud or WEB-GL based solution like VTK.JS [Kit18b] or ParaviewWEB [Kit18a] fail to provide the forecasters with required system responsiveness, which can be called “instant 3D”.

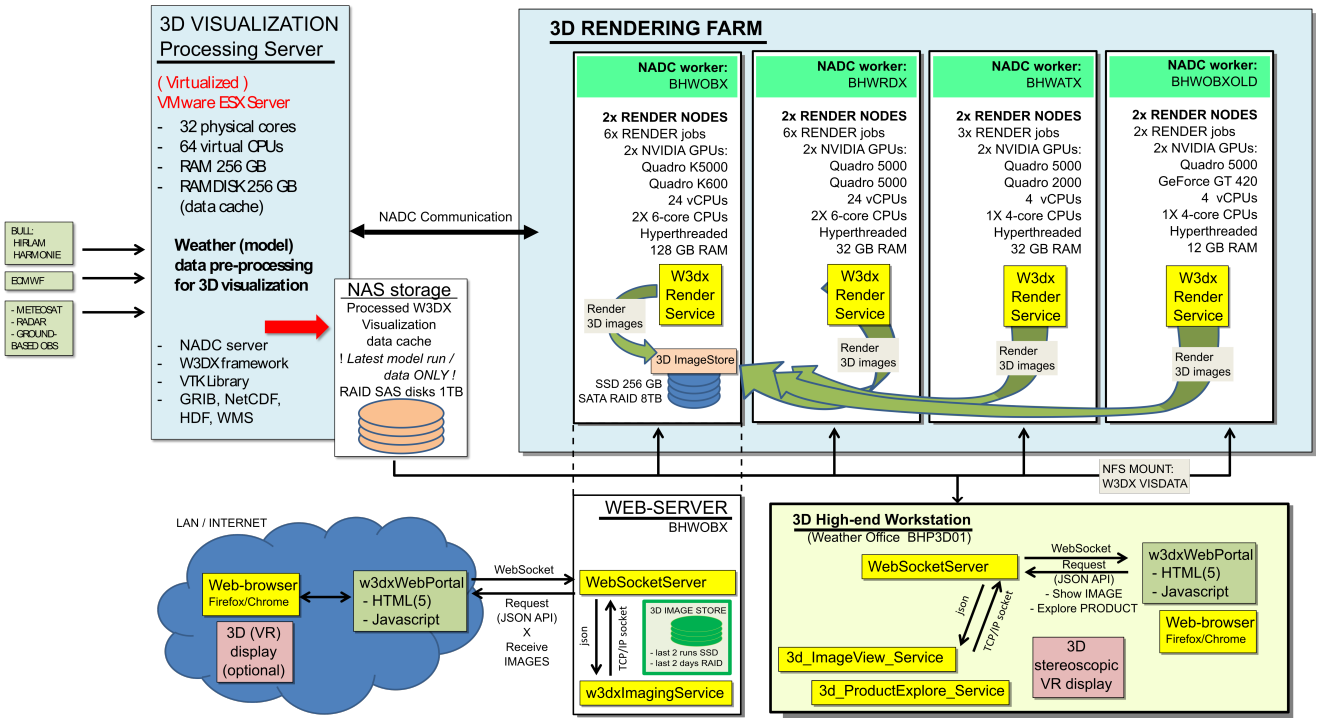

Figure 2: *W3DX 2.0 Visualization Framework: Architecture overview with main components: 3D Visualization Processing Server with NADC, 3D Rendering Farm, WEB-Server with W3DX Web-portal, high-end 3D/VR workstation(s).*

In late 2012 we installed our first generation interactive 3D meteo-visualization software (W3DX 1.0) in the Weather Forecasting Office. This tool already had a great track-record in meteo case-studies, VR demonstrations, supporting education and training [KvdNL*11]. Initially the first 3D graphics workstation went to the Briefing Room, which was instructive for demos, but later in 2013 we had to install the second workstation on the desk of the chief (guidance) meteorologist. Many forecasters were initially very interested in the new capabilities of interactive 3D visualization, combining NWP model data with satellite and radar observations, making cross-sections, etc. Several of the forecasters are still using the old system today. Nevertheless, in 2014 it became evident that interactive 3D visualization system cannot compete in an environment of 2D meteorological web-portals with pre-rendered images. Several issues were identified: (a) unfamiliar 3D user interface, (b) choosing a 3D product takes too much time, (c) navigation in time is slow, (d) 3D camera navigation is difficult.

In 2015 this resulted in a new project for developing operational user interfaces for 3D meteo-visualization. The motivation was to remove virtually every technical obstacle that would hinder use of 3D visualization products by operational forecasters. Therefore the Second generation Weather 3D eXplorer (W3DX 2.0) Visualization Framework has undergone an architectural re-design on the user-interface part (front-end) and NWP-model 3D post-processing part (back-end) to allow highly interactive 3D meteo-visualization in the web-portal with the possibility of providing VR experience directly in the browser by showing 3D stereoscopic images in full-screen mode (3D-top-bottom) as well as with the '1-click-away' feature to interactively explore, in VR, any of the pre-rendered visualization products from the web-portal. The decision to rigorously pre-render "everything" is important from the research and development perspective to study performance implications of different (remote) visualization and rendering solutions, with the ultimate goal to render on-demand, possibly in the GPU-graphics cloud in the future. We could not apply existing rendering farm solutions as these are aimed at motion pictures industry, engineering, and architecture where temporal navigation responsiveness is not critical. The following section will describe the main architecture components and the data-flow through the system.

2. W3DX 2.0 Visualization Framework

The back-end of W3DX 2.0 Visualization Framework is implemented using NADC processing suite [vdNvdVV17]. The NADC services operate on the dedicated 3D visualization processing server, see Figure 2. W3DX is built on top of Visualization Toolkit [SML06], [HMCA15]. We have developed extension modules for VTK in C++ and Python to handle the NWP model data and to do the 3D visualization processing and the product rendering more efficiently.

The front-end of W3DX 2.0 provides the W3DX web-portal for previewing of the visualization products and interaction with the W3DX services (websockets), written in Javascript and Python. Additionally 3D workstations can run the 3D-Image-View and the 3D-Product-Explore services so that the visualization products could be interactively explored on separate VR displays.

3D Visualization Processing Server

Several times a day the supercomputer delivers new NWP model data (forecast 0 to +48 hours) which are processed on the 3D visualization server. The processing is executed massively parallel for

many forecast-times simultaneously, see Figure 2 left. Additionally a number of the visualization processing techniques run in a multi-threaded mode. Vertical grid coordinates (geographic height in meters and in feet) and various derived meteorological fields are pre-computed and stored on the VTK structured-grid. For technical reasons to save a disk space and not to waste I/O resources we store the data as vtkImageData (.vti) files. Although the NWP models have curvilinear grids there is no need to store X,Y,Z coordinate for every grid-point. The horizontal coordinates are the same for every timestep of the given model. Only the vertical grid coordinates are dynamic, being different for every gridpoint. In the preparation phase, the NWP model grid is projected into a Cartesian coordinate space by using the North-pole stereographic cartographic projection. The projected horizontal grid together with the model orography information is stored into a grid-template file for a certain model and later used to reconstruct VTK structured-grid from the simple VTK-image-data (.vti) files. The geographic height of each grid point is in the data and it will provide the Z-coordinate. We often multiply the Z-axis with a certain scaling factor (10x, 20x, 50x or even 100x) depending on the visualization product definition.

The next processing step on the visualization server takes the VTK gridded data and creates 3D visualization pre-fabricates / products (VTK polydata output format: .vtp files) for later use during rendering. The .vtp files are compressed as well. At this stage we have all of VTK's 2D en 3D visualization techniques at our disposal. We had to implement a different colormapping and color-legenda to fulfill requirements of operational forecasters. The color-mapping in meteorology should use discrete, well defined, colors. The color-legenda is not rendered by VTK into the image, but superimposed during styling in the (HTML) canvas in the web-portal.

The W3DX visualization system uses VTK and offers the following types of 2D en 3D products: color-mapped iso-surfaces, ground-field maps with multiple layers of color-planes, vectors, and iso-lines, horizontal or vertical cross-sections also with multiple layers, observations (Meteosat and Radar images), streamlines, etc. We are using standard VTK visualization filters which we have improved by custom GLSL shaders and custom discrete color-mapping. Iso-surfaces of meteorological features such as clouds or jet-stream can be cut with a (vertical/horizontal) cross-section while showing the interior of these features in detail. For better user orientation we provide 2 vertical coordinates / guidelines: important pressure levels and altitude or flight levels. The (3D) visualization products are described using product-definition files in JSON format. These product definitions are pre-rendered for display in the W3DX Web-Portal, see Figures 1, 6. This enables instant rotation of the cross-sections and the cameras. When the user is working on a high-end 3D workstation the product definitions can be passed to the W3DX Product-Explorer Service for interactive VR exploration, see Figures 2, 7. It takes 1 click and about 1-2 seconds for the product to appear on a VR display.

3D Rendering Farm

The pre-computation of iso-surfaces, pre-slicing the 3D grid (3D products as VTK polydata) and only storing the data needed for the visualization products has a great performance benefit for the next step: image rendering of the products on the rendering farm, see Figure 2 right. To put this into a perspective, the data volume (per forecast) of the gridded data is hundreds of MBs (600MB - 1GB). The size of the VTK polydata (after compression of data and coordinates) is typically only tens of MBs. This amount of data needed for individual visualization product during the rendering job is something that even an ordinary 1GBit network with NFS shared storage services can handle. As we will show later in the text, this heterogeneous rendering system handles up to 17 parallel rendering jobs on 4 machines which access the (.vtp) 3D polydata on demand via a NFS data-share. Normally NADC would require to store all (3D) data needed for the (rendering) job into its working directory. This happens technically at (stage 2) as shown in Figure 3. This can however introduce costly data transfer operations. Testing revealed that the on-demand retrieval of next product data performed much better than gathering all the data into working directory at the beginning of the job. This has a pitfall: If the dependent data will not be on the NFS data-share when the render-job needs it, the job will fail. Therefore NADC prevents starting the render jobs until the dependent data is completely stored onto the (internal) NFS-share.

The 3D Rendering Farm with NADC processing framework has currently 4 render machines (hosts). Each machine (NADC worker) has 2x GPUs and 2x render nodes (processing units) assigned, for example: nadc-gpu0@bhwobx and nadc-gpu1@bhwobx. Depending on the GPU and the host performance we have assigned a maximum number of rendering jobs (processing slots). We make use of massive parallelism. We process (render) the 3D product-data of multiple independent forecast-times simultaneously on different render nodes. Each single render job renders thousands of 3D images, and it covers a complete set of visualization products for given a NWP model, for a single forecast-time, with all possible camera-views, and all predefined cross-section rotations. Views on the scene are rendered with different image resolutions.

Rendering Jobs in Detail

The NADC Job Manager differentiates between several job stages. The relevant stages of the render jobs are depicted in Figure 3. Every time the input data for the job becomes ready on the server (stage 1) the Job Manager will insert a new scheduled render job into the queue. The system has to wait until there is a processing unit with a free processing slot (stage 2).

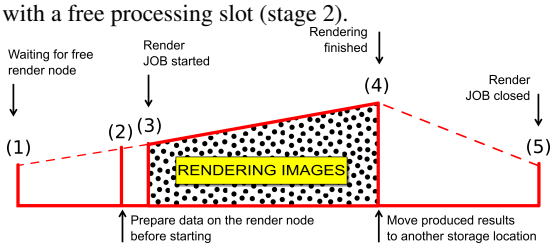

Figure 3: *Rendering jobs stages described*

When conditions for starting a new render job are in place the NADC Scheduler will select an available rendering node, setup a working directory for the render job, and upload the input data from the server to working dir. When this is ready (stage 3) the rendering job will be started. Images are rendered OFF-screen using fully GPU-accelerated EGL with high quality hardware anti-aliasing based on VTK 7.0 which supports the latest OpenGL and has a greatly improved support for the GPU shaders.

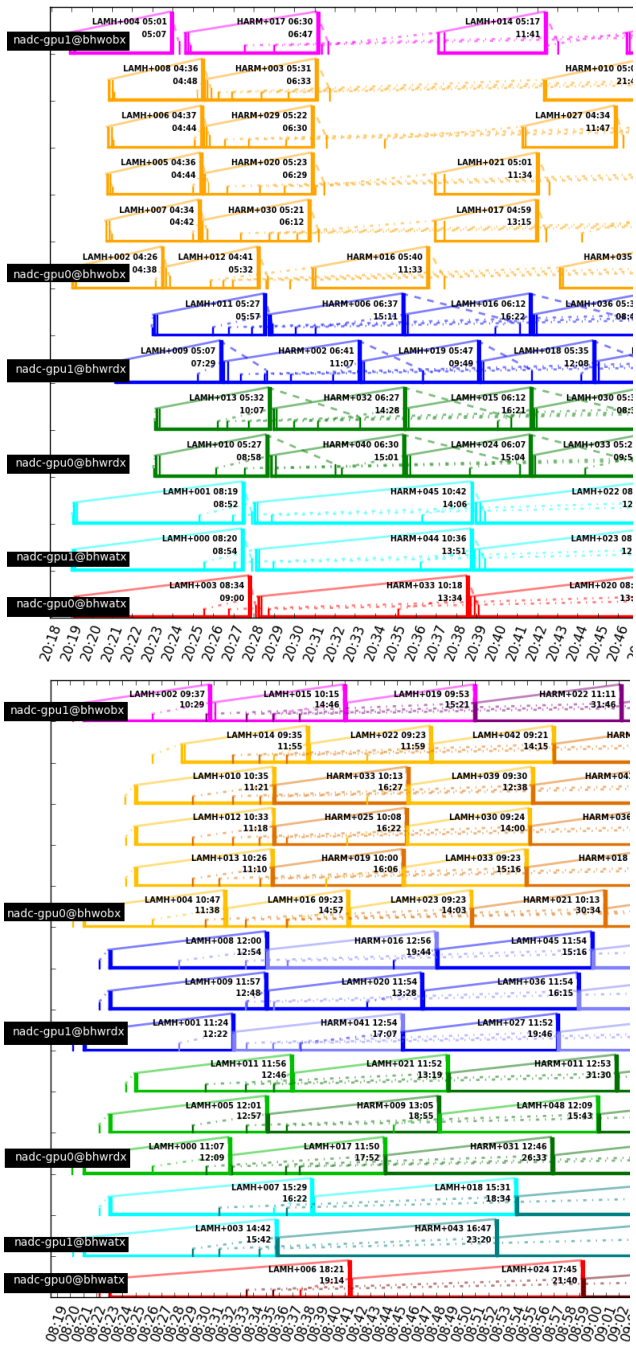

Figure 4: *Performance stats: Simultaneous processing of a small set of visualization products for 2 NWP models (HARMONIE, HIRLAM) at W3DX Rendering farm. Imbalanced situation (above) x optimized situation (below).*

Standard VTK modules grab the images into main-memory and produce JPEG files. Once all images (thousands) have been rendered and written into JPEG files in the TMP directory of the corresponding workernode in the RAMDISK then the render job puts all the rendered images into 1 image-tar file. At this point (stage 4) we consider the rendering as finished and the rendering resource (processing slot) is given back to NADC for rendering of a next request. Before the rendering job can be closed the output data (image-tar file) has to be moved by NADC to a different storage location. After this final step (stage 5) the NADC Scheduler will close the job.

The Rendering Scale and Scalability

The rendering is done upto +48 hours with 49 forecast-times for HARMONIE model (HARM) and 33 forecast-times for HIRLAM model (LAMH) (least frequent steps after +24 hours, +27, +30, +33, etc.). HARM model has a different set of visualization products than LAMH. Also HARM model products are more complicated to visualize. This explains the asymmetry in the render-job times between LAMH and HARM, see Figures 4, 5.

According to the product definitions from September 2017, the system is rendering 10506 HARMONIE images (1,6GB) and 9480 HIRLAM images (1,3GB) per forecast time step. With 4 model runs being processed per day, the system produces over 3.3 million images (= 4* (49*10506 + 33*9480) daily. Next to this we render visualization products containing satellite and radar images based on availability of the observations; Additionally per forecast-time another 1734 images with observations and HIRLAM data, and 2046 images with observations and HARMONIE data is produced.

After tuning the rendering farm we reached a stable situation where the processing continues without interruption for months. For stability reasons we have limited the number of processing slots to total of 17 = (5+1)+(3+3)+(1+2)+(2+0); [bhwobx + bhwrdx + bhwatx + bhwobxold]. This means we can do 17 render jobs in parallel, each with different forecast time and/or different NWP model. Due to the limited resources in the project we have built a rather heterogeneous rendering farm from several old recycled graphics workstations (in use sinds 2008, 2011, and 2014). The only exception is the BHWOBX machine which is a very powerful high-end graphics server with Quadro K5000 GPU and can easily render much more than 5 jobs simultaneously. As this machine is not only used as a rendering machine but also as a web-server and a file-server for the 3D data and the 3D image-store, we limited the this machine to 5+1 jobs, see Figure 2. In this way even under heavy load the web-portal remains responsive. For financial reasons we did not install a separate machine for the web-server, which would be a much better solution.

3. Results and Performance Optimizations

In Figure 4 we present some performance metrics of the 3D rendering farm. The first example shows a bad performance with hickups, gaps in availability of rendering resources, and delays in output data distribution. The second example shows a performance-optimized rendering farm where the new jobs start immediately when processing slots become available. In September 2017 we added extra 3D visualization products including 360-deg rotating cross-sections to the render set and the overall processing duration (two NWP models: LAMH and HARM) increased, see Figure 5.

During development and testing we have found out that disk I/O is critical for good overall performance of the system, especially the disk latency, and the read and write throughput. The current implementation uses the RAMDISK of each render node as working directory. All JPEG images are written to RAMDISK. Also creation of the image-bundle (image-tar file) happens on RAMDISK. When the render job is being closed, NADC will transport the resulting

Table 1: *MEAN TIMES per RESOLUTION aggregated over all visualization products: HARMONIE model (above), HIRLAM model (below). The total (mean average) time per 1 file contains: a) changing the camera-view b) rendering by GPU c) grabbing image from GPU to memory d) jpeg image encoding and writting image onto a filesystem.*

total [s]	imageEncode [s]	grabbing [s]	rendering [s]	cameraView [s]	imageResolution
0.396757	0.156725 (39.502%)	0.237645 (59.897%)	0.000500 (0.126%)	0.001886	res3DTBHD
0.202886	0.151590 (74.717%)	0.048869 (24.087%)	0.000528 (0.260%)	0.001899	resFullHD
0.073433	0.041056 (55.909%)	0.029767 (40.536%)	0.000535 (0.728%)	0.002076	resWeb800
0.072239	0.020764 (28.743%)	0.049198 (68.105%)	0.000516 (0.715%)	0.001761	resWeb540
0.041855	0.012330 (29.459%)	0.027088 (64.719%)	0.000547 (1.307%)	0.001890	resWeb400
0.035686	0.007030 (19.698%)	0.026109 (73.161%)	0.000546 (1.530%)	0.002002	resWeb280
0.241568	0.105522 (43.682%)	0.134411 (55.641%)	0.000339 (0.140%)	0.001296	res3DTBHD
0.139793	0.103398 (73.965%)	0.034776 (24.877%)	0.000341 (0.244%)	0.001278	resFullHD
0.054970	0.028577 (51.986%)	0.024751 (45.027%)	0.000346 (0.629%)	0.001297	resWeb800
0.058182	0.014358 (24.677%)	0.042284 (72.674%)	0.000351 (0.604%)	0.001190	resWeb540
0.032740	0.008538 (26.078%)	0.022563 (68.917%)	0.000361 (1.102%)	0.001278	resWeb400
0.027854	0.004787 (17.185%)	0.021445 (76.991%)	0.000366 (1.316%)	0.001256	resWeb280

dataset (image-tar file) to a storage location on the W3dx Imaging Server (BHWOBX). It became apparent that using internal RAID array inside the W3dx Imaging Server (BHWOBX) was too slow for uploading of the image-tar files and the rendering jobs where not closing rapidly.

The solution was to upload the image-tar files into the RAMDISK of the Imaging Server directly from the render nodes. NADC triggers the UnpackImages-job on the Imaging Server and the thousands of JPEG images are extracted from RAMDISK into 2 locations: (a) the primary image store on very fast SSD disc and (b) the secondary image store on SATA RAID array inside the same machine (BHWOBX). This way we minimize the time in between phases 4 and 5 of each render job in the rendering farm. Also the RAID disk resources in BHWOBX machine are spared.

Table 1 shows mean-average times during rendering multiple images for a complete product-set for two models for one forecast-time. For an optimal low-latency web-viewing we provide these resolutions: 800x600, 540x405, 400x300, 280x210. For fullscreen viewing the FullHD 1920x1080 and the 3D stereoscopic TOP-BOTTOM 1920x1080 are used. Table 1 also clearly indicates that pure rendering is extremely fast. The most time-consuming actions are: reading of the image-data back from the framebuffer of the GPU to the main memory (grabbing) and JPEG image CPU encoding. Both actions are provisioned by VTK OpenGL (standard) routines and the standard Linux JPEG libraries. During early development and testing in 2015/16 we compared image encoding performance of PNG and JPEG. Surprisingly the PNG format was extremely slow to encode and to write due to the files' large size. The only advantage of PNG was better visual clarity of images. JPEG quality was optimized based on user feedback.

It was not our primary goal to produce the 3D images with maximum performance. But rather to implement a working proof-of-concept system that satisfies high demands on the web-portal front-end responsiveness. But if it will be required in the future to dramatically shorten the overall rendering time (now it is about 2 hours for 2 complete NWP model runs), as shown in Figure 5, it will be important to optimize the image-encoding and the image read-back from GPU. The rendering process spends now more than 90% of time on these 2 actions. Preferably the encoding should be done on GPU itself. This will lead to dramatic performance improvements.

Table 2: *Visualization pipeline setup duration for a certain visualization product from the total product-sets. This relates to one rendering job (a certain forcast-time) for the respective HARMONIE (HARM) or HIRLAM (LAMH) models. The measured times precede the actual image rendering and production of image files.*

Setup [sec]	visProdId
2.218000	HARM-CloudsWithJet
0.646000	HARM-Clouds
0.728000	HARM-Clouds-TCC
3.038000	HARM-JetStream
2.232000	HARM-Rain
5.899000	HARM-CrossSection-DeBilt-Wind
6.508000	HARM-CrossSection-DeBilt-Clouds-Precip
6.490000	HARM-CrossSection-DeBilt-PrecipType-Clouds
5.427000	HARM-CrossSection-DeBilt-VelocityW
6.061000	HARM-CrossSection-DeBilt-ThetaW-clouds
13.24600	HARM-CrossSection-DeBilt-Temp
1.391000	LAMH-CloudsWithJet
0.849000	LAMH-Clouds
0.427000	LAMH-Clouds-TCC
1.385000	LAMH-JetStream
5.270000	LAMH-CrossSection-DeBilt-Wind
4.155000	LAMH-CrossSection-DeBilt-Clouds-Precip
5.017000	LAMH-CrossSection-DeBilt-ThetaW-clouds
2.805000	LAMH-CrossSection-DeBilt-ThetaW
3.995000	LAMH-CrossSection-DeBilt-Temp

Table 2 shows times needed to prepare the 3D visualization pipeline and the 3D scene for rendering of a certain product. The setup includes reading the 3D product-data (in an optimized form) from disk, building the VTK visualization pipeline, and shipping the 3D geometries to the GPU for rendering. The setup times are in the order of seconds but there can be outliers of more than 10 seconds. This is something that has to be done for every new forecast-time that the system should render.

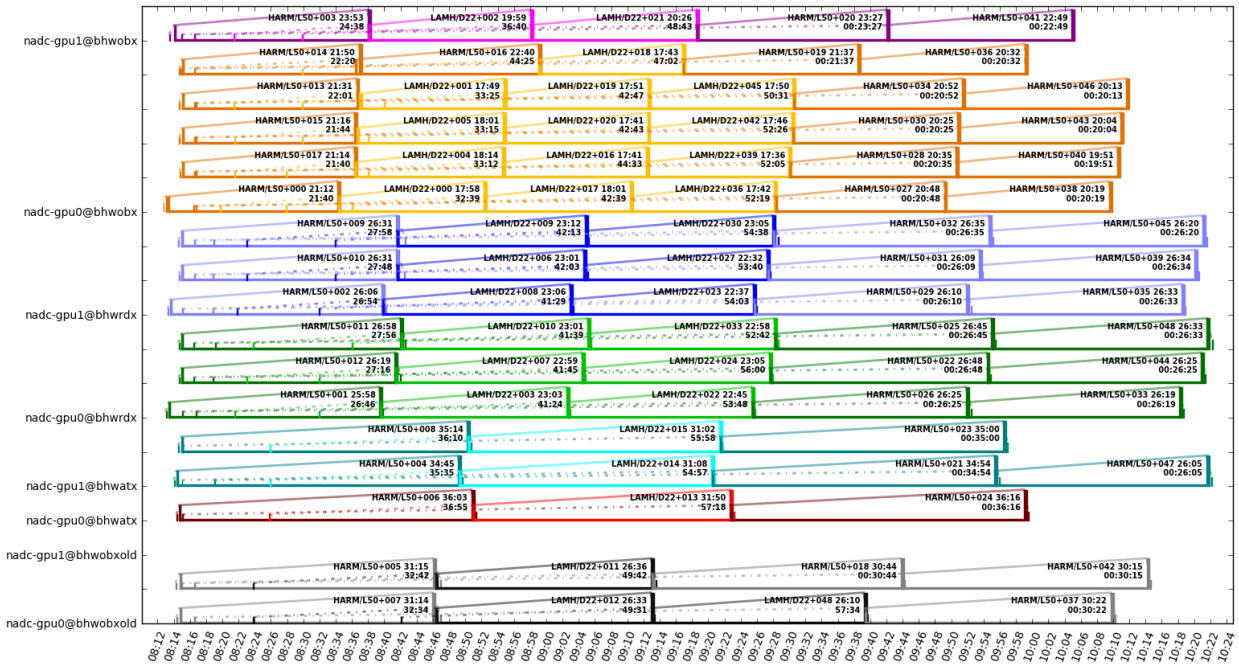

Figure 5: *Rendering Performance Stats: Simultaneous heavy load processing of a large sets of visualization products for 2 NWP models.*

Basically the setup latencies from above are the primary reason why remote visualization on a server or local visualization inside a web-browser using WEB-GL would deliver unresponsive behavior not suitable for operational use. The forecasters demand almost immediate response. Waiting a couple of seconds for visualization of a next forecast-time is considered unacceptable.

4. Discussion on Implementation and Results

During the early developments in 2015 we were using the development git-version of VTK 6.0 which was suffering from a poor and limited OpenGL (GPU) support. At that time the only way to render off-screen was the slow Mesa OpenGL with poor quality anti-aliasing. Therefore we tested various ways to do GPU accelared on-screen 3D rendering with VTK without overlapping windows under X11-windows under Linux. Finally in 2016 the VTK 7.0 has been released with greatly improved OpenGL support [HMCA15] and availability to use GPU-accelarated EGL [LGA15], [Mes16]. This resolved the poor off-screen rendering performance and high quality GPU anti-aliasing.

The POC implementation intended to provide maximum responsiveness to the user (forecaster) in the W3DX web-portal. The forecasters visually study the (3D) dynamics of the meteorological features. They like to navigate smoothly and quickly through the forecast-time while observing the (3D) meteo-dynamics.

Using various visualization layouts we can put different 2D and 3D products together. Depending on the page layout the portal decides on the optimal resolution. In Figure 6 the web-portal shows 6 views simultaneously. In this layout all images have 540x405 pixels. The images are pre-rendered with high-quality anti-aliasing at this exact resolution. To keep the network load on the server at minimum and to ensure the highest quality images we don't downscale to smaller images from 1920x1080 (HD) resolution in the browser. To maximize the web-portal responsiveness we even use 2 types of image-storage: (a) SSD for 2 latest model runs, (b) SATA RAID upto 24 hours old runs. In this example with 6-images layout, when the user advances to a next forecast-time (or rotates the camera(s)) the new images have to be loaded. In the case of latest model-runs (SSD) this action takes about 15ms = 6 x 2,5ms; with JPEG-file size around 50-60kB and individual reading times 1-4ms. When the user want to see what the 24 hours-old run computed for today (SATA RAID) this same layout takes about 250ms = 6x42ms with individual reading times 11-72ms. When looking on the fullscreen HD images in the web-portal for the SSD the average reading time is about 4ms and the SATA RAID is about 17ms. These are very interactive refresh rates that cannot be challenged by a generic remote visualization system (for example ParaviewWEB [Kit18a] or IDV in the cloud [Fis17]).

We do however use Paraview during development and designing of new products. The possibility inside the web-portal to combine many different 3D and 2D products, different meteo-features, different camera-views, or different NWP model versions into one layout makes the 3D rendering farm solution the only practical and scalable solution for time-constraint working environments such as operational weather forecasting. Valid questions in this context are: What should be (pre-)rendered and when?

5. Conclusions and Outlook

In this paper we have presented a new framework for meteorological 3D visualization that levarages 3D rendering farm to enable highly responsive visualization in a web-portal. Figure 7 shows the W3DX 2.0 system installed in the Weather Forecasting Office. The W3DX web-portal is available to every forecaster.

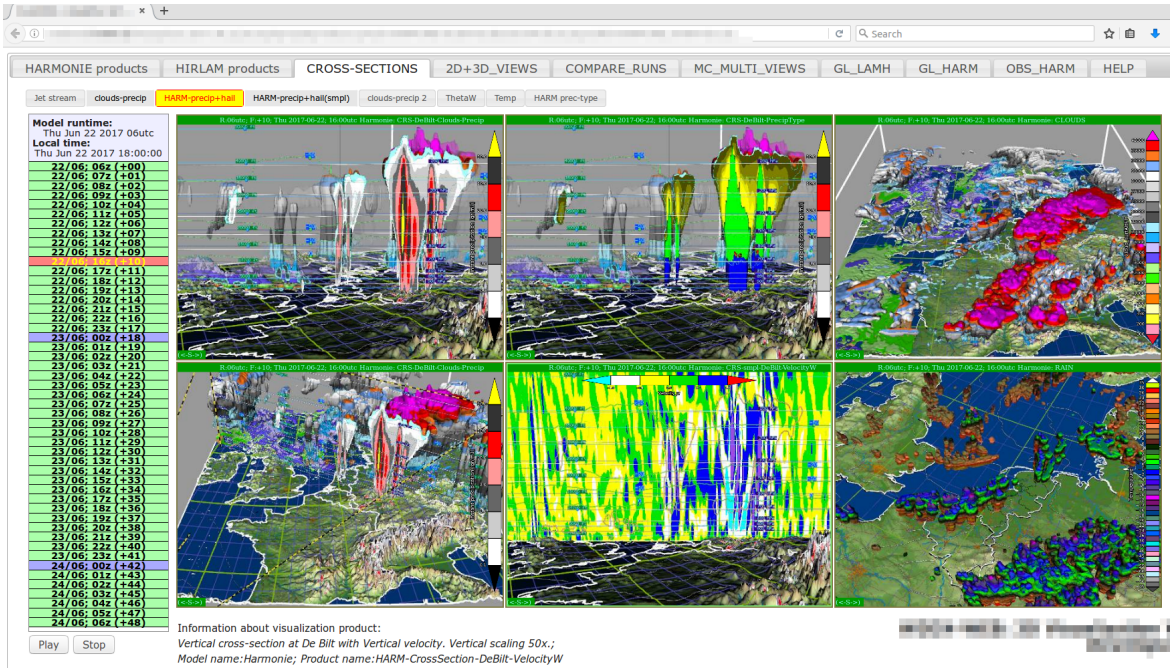

Figure 6: *A thunderstorm explored using W3DX Web-Portal showing simultaneously different weather modalities: 3D clouds, cross-sections containing various fields: precipitation intensity (left upper + below), dominant precipitation types (center upper) - snow, graupel/hail, rain, freezing rain, vertical wind component (center below) showing convection or down-bursts. 3D clouds colormapped with height (right upper) showing covective cloud-tops, rain colormapped with temperature (right below) showing super-cooled/freezing rain with danger of icing.*

The chief (guidance) forecaster also has a 3D visualization workstation on her/his desk and can take a closer look on any of the product previews from the web-portal on a 3D HD television. Also the briefing room of the Weather Forecasting Office is equipped with a large 3D television to show any of the 3D images and to explore the 3D products interactively. In the interactive mode the 3D products can eventually be modified, cross-sections can be moved to another location and rotated as needed. Wearing and using of the 3D glasses for the VR-effect is optional. Some of the forecasters find it sufficient to use the motion parallax and rotate the camera-view for better spatial understanding.

In future work we will include training of operational forecasters and defining useful visualization products in cooperation with the forecasters. We expect that making this system operational will include off-loading of the rendering into a 3D rendering cloud. It will remain important that the visualization processing takes place close to the data and that the image data moves through the system efficiently. Due to the advances in the NVIDIA GPU driven cloud solutions it might be possible to run interactive 3D visualization sessions remotely, as recently presented with IDV in the cloud [Fis17]. We expect great challenges with respect to responsiveness during opening of new (3D) visualization products and advancing the time-step (forecast-time). There will be also limits on how many concurrent users could use such remote rendering facility and limits to how many 3D products one user can request simultaneously.

Another interesting option will be to investigate 3D rendering inside a web-browser using WEB-GL [Par12] Javascript libraries:

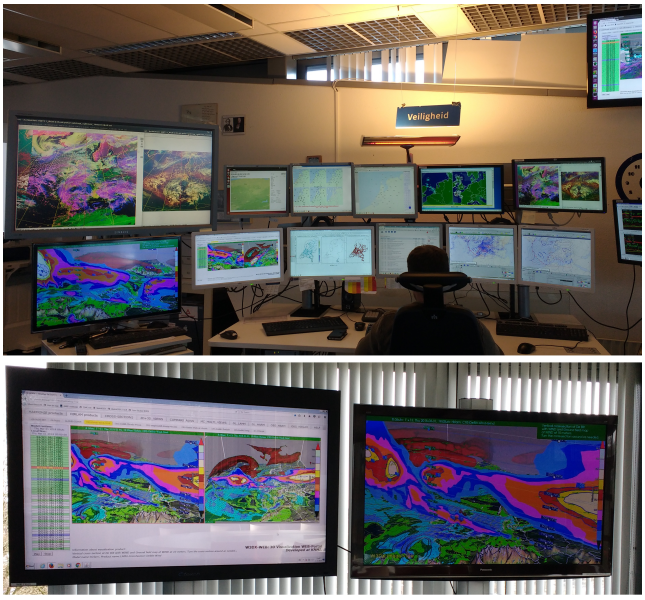

Figure 7: *The W3DX Web-portal in the Weather Office: Workdesk of the forecaster (image above), Weather briefing room (below), the 3D television next to the screen with the browser for VR experience.*

Threejs [Dir13], VTK.JS [Kit18b], and Cesium [CF18]. This will however suffer from significant setup/startup times because the 3D visualization products data has to be off-loaded from the server to the (web-)client. In our case the size of the 3D geometry data

of our typical 3D products varies between 40MB - 200MB per scene/frame/timestep. Once the 3D scene data is on the client the GPU-accelarated rendering is very fast. A practical usage of such approach in meteo-education could be exploring of one 3D visualization product and one forecast-time at the time.

Acknowledgments

We would like thank the Royal Netherlands Meteorological Institute (KNMI) for supporting this research and allowing us to test this experimental system in the Weather Office. Many colleaques from several KNMI departements have contributed to the overall system: NWP model researchers, data technology developers, production engineers, and operational weather forecasters;
a special thanks to Sander Tijm and Toon Moene for providing operational NWP model data and Frans Debie, Marcel Molendijk, Jos Diepeveen, other forecasters for helping us with visualization product definitions, testing the system and providing feedback.

References

[AGL05] AHRENS J., GEVECI B., LAW C.: Paraview: An end-user tool for large data visualization. In *Visualization Handbook* (01 2005). 2

[BAA*17] BENGTSSON L., ANDRAE U., ASPELIEN T., BATRAK Y., CALVO J., DE ROOY W., ET AL.: The harmonie - arome model configuration in the aladin - hirlam nwp system. *Monthly Weather Review 145*, 5 (2017), 1919–1935. URL: https://doi.org/10.1175/MWR-D-16-0417.1. 2

[CF18] CESIUM-FORUM: Cesium javascript webgl library, 2018. http://cesiumjs.org/. 8

[CMNR07] CLYNE J., MININNI P., NORTON A., RAST M.: Interactive desktop analysis of high resolution simulations: application to turbulent plume dynamics and current sheet formation. *New Journal of Physics 9*, 8 (2007), 301. URL: http://stacks.iop.org/1367-2630/9/i=8/a=301. 2

[Dir13] DIRKSEN J.: *Learning Three.js: The JavaScript 3D Library for WebGL*. UK: Packt Publishing, 2013. URL: http://threejs.org/. 8

[Fis17] FISHER W.: Cloud-based data-proximate visualization and analysis. *Geophysical Research Abstracts EGU General Assembly 19*, EGU2017-9011 (2017). 2, 7, 8

[HAF*96] HIBBARD W. L., ANDERSON J., FOSTER I., PAUL B. E., JACOB R., SCHAFER C., TYREE M. K.: Exploring coupled atmosphere-ocean models using vis5d. *The International Journal of Supercomputer Applications and High Performance Computing 10*, 2-3 (1996), 211–222. doi:10.1177/109434209601000208. 2

[HBR*14] HELBIG C., BAUER H.-S., RINK K., WULFMEYER V., FRANK M., KOLDITZ O.: Concept and workflow for 3d visualization of atmospheric data in a virtual reality environment for analytical approaches. *Environmental Earth Sciences 72*, 10 (Nov 2014), 3767–3780. doi:10.1007/s12665-014-3136-6. 2

[HC04] HOFFMAN R., COFFEY J.: Weather forecasting and the principles of complex cognitive systems. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (09 2004), vol. 48. 2

[Hib05] HIBBARD W. L.: Vis5d, cave5d, and visad. In *Visualization Handbook* (2005), Hansen C. D., Johnson C., (Eds.), Academic Press, pp. 673–688. 2

[HMCA15] HANWELL M. D., MARTIN K. M., CHAUDHARY A., AVILA L. S.: The visualization toolkit (vtk): Rewriting the rendering code for modern graphics cards. *SoftwareX 1-2* (2015), 9 – 12. doi:https://doi.org/10.1016/j.softx.2015.04.001. 2, 3, 7

[HSUB89] HIBBARD W. L., SANTEK D., UCCELLINI L., BRILL K.: Application of the 4-d mcidas to a model diagnostic study of the presidents day cyclone. *Bull. Amer. Meteor. Soc. 70* (1989), 1394–1403. 2

[KDvdN16] KOUTEK M., DEBIE F., VAN DER NEUT I.: 3d exploration of meteorological data: Facing the challenges of operational forecasters. In *EGU General Assembly Conference Abstracts* (Apr. 2016), vol. 18. 2

[Kit18a] KITWARE: The paraview visualizer: Scientific visualization on the web using a paraview backend., 2018. https://kitware.github.io/visualizer/docs/. 2, 7

[Kit18b] KITWARE: Vtk.js: A rendering library made for scientific visualization on the web, 2018. https://kitware.github.io/vtk-js/docs/. 2, 8

[KSH*98] KOPPERT H. J., SCHRÖDER F., HERGENRÖTHER E., LUX M., TREMBILSKI A.: 3d visualisation in daily operation at the dwd. In *Proceedings of the 6th ECMWF Workshop on Meteorological Operational Systems, 17-21 November 1997, Reading, England* (1998), pp. 101–125. 2

[KvdNL*11] KOUTEK M., VAN DER NEUT I., LEMCKE K., ET AL.: Exploration of severe weather events in virtual reality environments: Lessons learned from interactive 3d visualization of meteorological models. In *European conference on applications of meteorology EMS Annual Meeting, Berlin* (2011). URL: http://projects.knmi.nl/w3dx. 2, 3

[LGA15] LIPSA D., GEVECI B., AYACHIT U.: Offscreen rendering through the native platform interface egl, kitware, 2015. https://blog.kitware.com/off-screen-rendering/-through-the-native-platform-interface-egl. 7

[Mes16] MESSMER P.: Egl eye: Opengl visualization without an x server, nvidia, 2016. https://devblogs.nvidia.com/egl-eye-opengl-visualization-without-x-server. 7

[MM07] MURRAY D., MCWHIRTER J.: Evolving idv - creating better tools for the community. *23th Conference on International Interactive Information and Processing Systems (IIPS) for Meteorology,Oceanography, and Hydrology, 15-18 January 2007, San Antonio, TX, American Meteorological Society, 3B.5* (2007). 2

[MMS00] MCCASLIN P. T., MCDONALD P. A., SZOKE E. J.: 3d visualization development at noaa forecast systems laboratory. *ACM SIGGRAPH Comput. Graph. 34* (2000), 41–44. 2

[Par12] PARISI T.: *Webgl Up and Running*. Sebastopol: Oreilly & Associates Inc., 2012. URL: www.khronos.org/webgl/wiki. 8

[PdVT*18] PLIEGER M., DE VREEDE E., TERPSTRA M., VAN MOOSEL W., VAN DE VEGTE J.: Adaguc open source visualization utilized in the knmi geoweb project. *Geophysical Research Abstracts EGU General Assembly 20*, EGU2018-17245 (2018). 2

[RBK17] RINK K., BILKE L., KOLDITZ O.: Setting up virtual geographic environments in unity. In *Workshop on Visualisation in Environmental Sciences (EnvirVis)* (2017), Rink K., Middel A., Zeckzer D., Bujack R., (Eds.), The Eurographics Association. doi:10.2312/envirvis.20171096. 2

[RKSW15] RAUTENHAUS M., KERN M., SCHAFLER A., WESTERMANN R.: Three-dimensional visualization of ensemble weather forecasts part 1: The visualization tool met 3d. *Geoscientific Model Development 8*, 7 (2015), 2329–2353. URL: https://www.geosci-model-dev.net/8/2329/2015/. 2

[SML06] SCHROEDER W., MARTIN K., LORENSEN B.: *The Visualization Toolkit (4th ed.)*. Kitware, 2006. 3

[vDLS02] VAN DAM A., LAIDLAW D. H., SIMPSON R. M.: Experiments in immersive virtual reality for scientific visualization. *Computers & Graphics 26*, 4 (2002), 535–555. URL: https://doi.org/10.1016/S0097-8493(02)00113-9. 2

[vdNvdVV17] VAN DER NEUT I., VAN DE VEGTE J., VERHOEF H.: The netherlands atmospheric data center procession suite (nadc), 2017. http://projects.knmi.nl/w3dx/inprogress/references/Product_NADC_v5.pdf. 3