In-Situ Processing in Climate Science

Niklas Röber(✉) and Jan Frederik Engels

German Climate Computing Center, Hamburg, Germany
{roeber,engels}@dkrz.de

Abstract. With climate simulations and earth observations, earth system sciences belong to the most data intensive scientific disciplines, and the rate at which the data is produced increases continuously. Current models supporting a higher complexity paired with an increased resolution produce more and more data that needs to be analyzed and understood. The development of alternatives to the classic post processing/visualization pipeline are therefore mandatory and discussed within this paper, with a strong focus on in-situ visualization and in-situ data processing. Although the work described here is work in progress, large parts are already implemented and tested and on the verge to be deployed in production mode.

1 Visualization in Climate Science

The output generated by current climate simulations is increasing both in size, as well as in complexity. Both aspects pose equal challenges for the visualization and an ideally interactive visual analysis of the simulated data. The increase in complexity is due to a maturing of models that are able to better describe the intricacies of the climate system, while the gain in size is a direct result of finer spatial and temporal resolutions.

ICON, the **ICO**sahedral **N**on-hydrostatic model, that is jointly developed by the Max Planck Institute for Meteorology (MPI-M) and the German Weather Service (DWD), is a framework based on an icosahedral grid with an equal area projection, on which data sets are sampled via primal triangular cells, dual hexagonal cells and hybrid quadrilateral cells [1]. Figure 1 shows the horizontal layout of the ICON grid, visualizing the relationship between cell (triangle) and point (hexagon) data. The vertical layout is a rectilinear grid, that is sampled more densely close to the Earth's/Ocean's surface. ICON – though unstructured – has several advantages over other grids that are regularly used in climate science: is has no computational poles, it allows for an easy refinement in local areas and it provides a simplified coupling between its oceanic, atmospheric and land components. Over the last years, ICON was extended to permit large eddy simulations at cloud resolving resolutions in a regional setup as part of the HD(CP)^2 project[1] [2] to advance the understanding of clouds, cloud building and precipitation processes. Although the data produced was quite large (22 million/3.5 billion

[1] HD(CP)^2 – **H**ighHigh-**D**efinition **C**louds and **P**recipitation to advance **C**limate **P**rediction.

M. Weiland et al. (Eds.): ISC 2019 Workshops, LNCS 11887, pp. 612–622, 2019.
https://doi.org/10.1007/978-3-030-34356-9_46

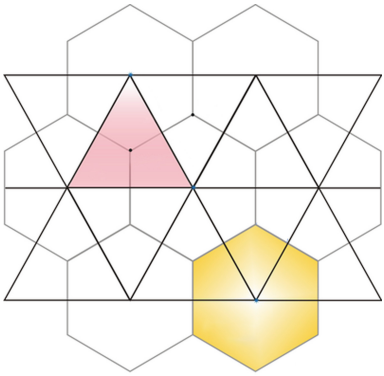

Fig. 1. Horizontal ICON grid layout showing triangles (cell data) and hexagons (point data).

cells 2D/3D), a classic post visualization approach using ParaView employing a parallel processing/visualization setup on several fat nodes was still possible.

Within the recently started EU funded project of ESiWACE2[2], the spatial – and the temporal – resolution will be further refined down to 1.25 km globally, resulting in approximately 360 million cells per level, and – depending on the number of levels – around 30 to 60 billion cells in 3D, per variable and time step. In order to explore, and actually be able to access such data, let alone writing the data to disk, other workflows than the currently employed post visualization are necessary. Examples include in-situ visualization in its many forms and in-situ compression/transformation, which reorders and possibly also compresses the data to make it accessible within a modified post visualization pipeline. This paper illuminates these approaches from a climate science perspective, thereby focusing explicitly on climate science visualization needs. It also discusses some of the initial implementations and highlights some first results.

2 In-Situ Visualization

The idea of in-situ visualization dates back to the golden era of coprocessing in the 1990s [3]. As a buzzword in HPC, the term *in-situ* is currently almost as popular as *data avalanche* or *I/O bottleneck*, and reverberated through scientific conferences and papers for years [3,4]. However, the majority of users still tried to avoid it united in the hope that faster hardware could remedy the problems before the need to resort to an in-situ visualization approach would become immanent. So far, also the data analysis and visualization at DKRZ was entirely based on the classic post visualization workflow, but this is – driven by projects such as ESiWACE2 – about to change. Only one simulated day with 30 min

[2] ESiWACE2 – Centre of **E**xcellence in **Si**mulation of **W**eather **A**nd **C**limate in **E**urope.

output, 75 levels and eight 3D plus twenty 2D variables – there are of course many more variables in the model that would be worth looking at – would accumulate to ≈43 TB (single precision), which is quite a bit, given the fact that such weather simulations often run for several days, weeks, even months.

Several software packages, such as Visit [5] and ParaView [6,7], already provide in-situ visualization capabilities. Other exascale visualization initiatives, such as ALPINE/Ascent [8] or SENSEI [9], directly build upon these tools and extend their functionality. Due to the great familiarity and happiness with ParaView, we at DKRZ have experimented so far only with Catalyst, a VTK-based and ParaView bound in-situ visualization framework [4,10]. ParaView is thereby employed to generate a Python script, that later drives the in-situ processing to create standard visualizations, to threshold and write out reduced data sets, and to create a CINEMA database. The connection to the ICON model is implemented using a so called Catalyst adaptor, compare also with Fig. 2, that handles the data transfer from ICON (FORTRAN) to ParaView/Catalyst (C++). As use cases, we see the following applications/data flows:

- Data reduction
- Verifying the simulation during run time
- Generation of data quicklooks and previews
- Feature detection, extraction and tracking

Once the data is on the C++ side of the adaptor, the data can flow in multiple directions, as are outlined in Fig. 2. Catalyst already supports a number of those applications and comes with a variety of examples. Initially we planned to base our in-situ developments onto the in-situ implementation developed by the MPAS[3] group [11]. However, as the code was quite complex and would have required a lot of work to be customized for ICON, we decided to start the development of a new Catalyst adaptor that is directly tailored to the ICON model from scratch. This proved to be the right decision, as only a few hundred lines of FORTRAN and C++ code, along with some minor modifications within the ICON code itself, already allowed us to run first in-situ visualization experiments. In-situ visualization can be performed either loosely or tightly coupled, that is on dedicated visualizations nodes, or on the same compute nodes on which the simulation is run. We have not yet experimented with different configurations, and so far have only used the tightly coupled setup with an equal number of visualization/simulation processes per node. The data, grid information as well as actual data variables, are transferred from FORTRAN to C++ as zero copy arrays.

The maximum number of nodes that we used so far for parallel simulation/in-situ processing is 540 nodes with 4320 MPI processes, i.e. 1/6 of our HPC cluster. ICON atmosphere was used in a global setup with 2.5 km horizontal resolution and 75 height levels, thereby thresholding two 3D variables *liquid cloud water* and *cloud ice*, which were written out to disk. A threshold of 1.0×10^{-7} kg kg^{-1} (in kilogram water/ice per kilogram air) was applied to discard the *empty* cells,

[3] MPAS – **M**odel for **P**rediction **A**cross **S**cales.

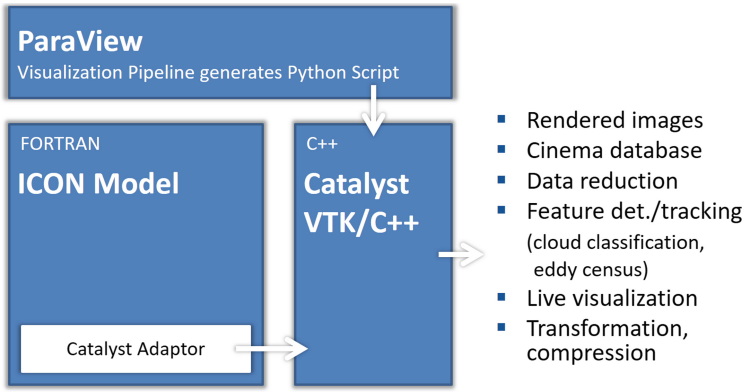

Fig. 2. In-Situ visualization/processing pipeline using ParaView/Catalyst.

and save only those that are above this threshold. This resulted in a mesh reduction from 6.5 billion cells down to ≈150 million cells per 3D variable, which can now be handled easily in a classic post analysis/visualization workflow. Figure 3 shows a visualization of both variables along with an annotation of the Earth's orography. Additionally, a standard visualization (single snapshot per time step) was created, along with a CINEMA database to create quicklooks and previews of the data.

The additional time required to initialize and actually do the in-situ visualization is almost neglectable in respect to the time required to perform the simulation itself. The initialization of the model, as described in the run above, takes 71 s, from which Catalyst needs 6.1 s (≈11%). The first workload, i.e. the transfer of the grid data and Catalyst initialization takes an additional 1.6 s. The model needs an average of about 408 s (total: 408.027 s) to advance the model one time step, from which Catalyst uses on average 0.1 s (max 0.8 s) (total: 12.000 s (≈3%)) to create a standard visualization (single image) and to write out the two thresholded 3D variables.

CINEMA is a useful extension for ParaView developed by the Los Alamos National Laboratory and allows an image-based analysis and visualization of large data sets by creating multiple views of the data per timestep and storing these images together with a data description. Several applications exist to efficiently access this CINEMA database, including a web based viewer [12]. But the evaluation of CINEMA is unfortunately still incomplete, as several issues exist that are also documented on Kitware's Github website[4]. Although the new SPEC-D CINEMA database format is formally implemented, currently only SPEC-A databases are written out and have at the time of writing still to be manually converted into a SPEC-D database to be consistent with the current CINEMA display tools[5]. Nevertheless, once working, CINEMA will be a great

[4] https://gitlab.kitware.com/paraview/paraview/issues/.
[5] https://cinemascience.github.io/downloads.html.

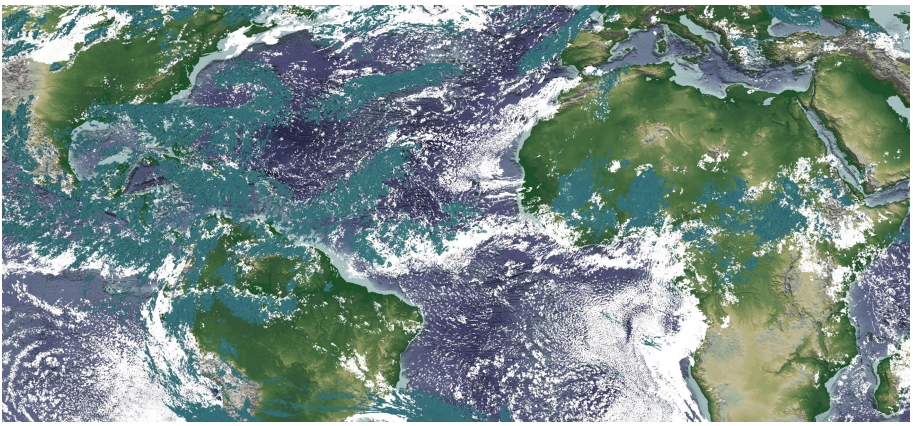

Fig. 3. Catalyst extracted 3D cloud ice (turquoise) and 3D cloud water (white) from a 2.5 km global ICON atmosphere simulation.

addition, as it allows scientists to quickly browse through large piles of data (images) in the search for the proper data file to be analyzed in more detail. Other researchers have also utilized the CINEMA database to perform a feature tracking of mesoscale ocean eddies using contours and moments directly on the image data [13].

Another direct advantage of using ParaView/Catalyst as in-situ framework is the possibility to visualize the simulated data live while the simulation still runs on the supercomputer. This allows to precisely track and supervise the progress of the simulation, and to abort in cases of errors. However, in practice, this feature will probably only be used in specific setups and for long simulations, as a batch scheduler maintains the processing of the simulations.

2.1 Feature Detection and Tracking

In-situ processing does not necessarily need to be limited to the generation of images and the storage of reduced data sets alone, but can easily be extended to a detection, extraction and tracking of certain interesting features or structures. A popular object of study in the atmospheric sciences are clouds. Meteorologists are especially interested in their formation, as well as development and evolution over time, i.e. the transition from one type of cloud into another. Clouds and the various cloud types are listed and described in the cloud atlas[6] and can be characterized using boundary conditions defined through specific levels of cloud water/ice, pressure, temperature, humidity, rain and upward wind velocity, as are identified by the International Satellite Cloud Climatology Project ISCCP [14]. Figure 4 shows a visualization of an in-situ cloud classification based on a regional (Germany centered) simulation at cloud resolving scales at 156 m

[6] http://www.wolken-online.de/wolkenatlas.htm.

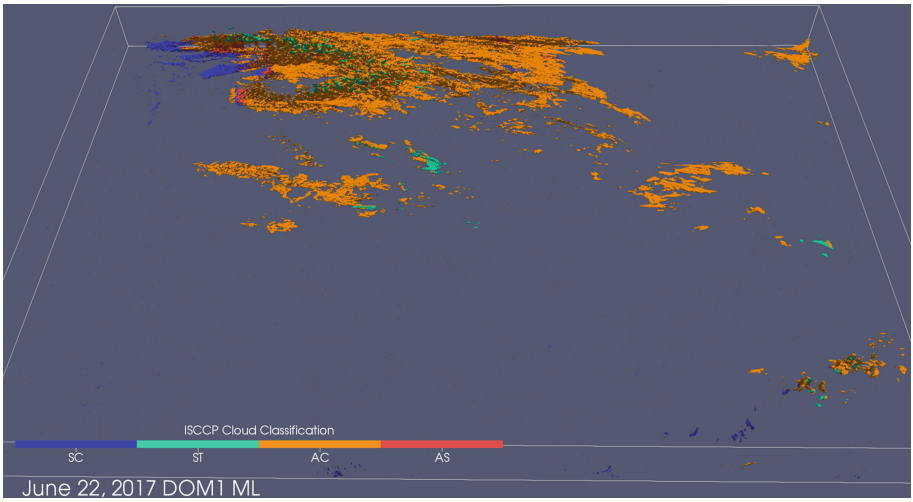

Fig. 4. ISCCP in-situ cloud classification using HD(CP)^2 data [2] with stratus (ST), stratocumulus (SC), sltostratus (AS) and sltocumulus (AC) clouds.

from within the HD(CP)^2 project [2]. It shows four different cloud types over northern Germany: Stratus (ST), Stratocumulus (SC), Altostratus (AS) and Altocumulus (AC). This cloud classification is thresholded, i.e. the empty cells (no clouds) are discarded, and efficiently stored as VTI (VTK Integer Array) file series on disk, to be later loaded and displayed in ParaView.

Another example that we have not yet implemented, but which has been shown by others to be beneficial [11], is an in-situ eddy census from a high resolution ocean simulation, ideally accompanied with the extraction and display of additional quantitative information, such as the size, duration, speed and movement direction of eddies, as well as how much energy and mass they transport.

3 Progressive Data Visualization

One of the drawbacks of an in-situ visualization/processing approach is the indispensable need of a priori knowledge to extract and visualize the right features in the simulated data. Without the domain knowledge where to find interesting features, and which isolevels and/or thresholds to choose, an in-situ visualization is likely to fail. An iterative approach is of course possible, but both time and labour intensive. The finer spatial and temporal resolutions of large scale simulations, also exhibit new – possibly previously unresolved – processes and correlations within the data. The thresholds and isolevels used in simulations at lower resolution can provide guidance, but are probably not a perfect fit at higher resolutions. To correctly find and visualize those new features and structures, one needs to work on the actual data in its original resolution, but the data needs to be transformed and reordered to make the data accessible. To achieve

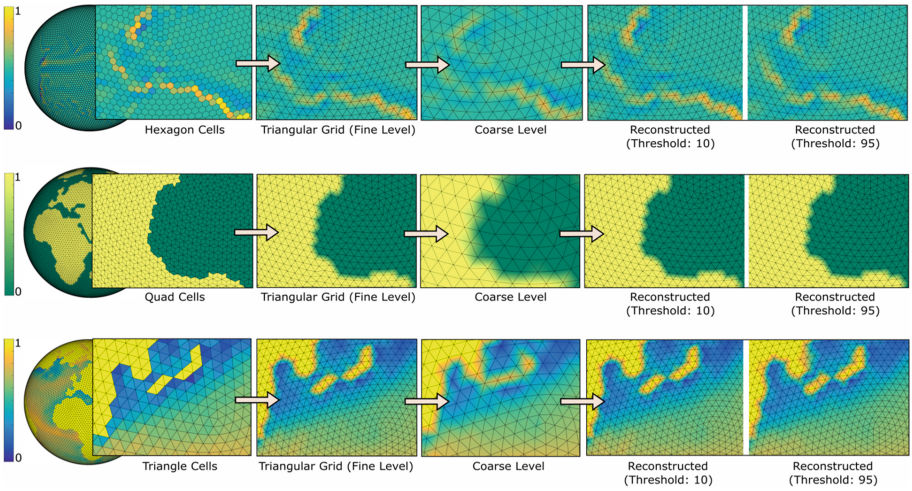

Fig. 5. Exemplary application of DHWT on a low resolution ICON ocean data set. Centroids of hexagons (top), edge midpoints in one direction (middle), centroids of triangles (bottom). Each row (left to right) shows original data, conversion to a triangular grid via icosahedral maps, applying DHWT on the converted grid to obtain coarse data, and reconstructed data using two different thresholds: 10% discarded/90% preserved and 95% discarded/5% preserved.

this, a progressive data visualization approach based on a level-of-detail (LoD) rendering is required. In here, the data is decomposed into different resolutions (LoD) and possibly also compressed before it is written out to disk in a way that facilitates a later interactive access. After the data has been written to disk, a special visualization application that supports progressive LoD rendering is used for a classic user-driven post visualization of the data [15]. Such an application accesses data in an out-of-core fashion, and only the data that is relevant to the current level of detail and which is also contained in the current view frustum is fetched, visualized and displayed.

3.1 Wavelet Decomposition and Compression

A classic tool for a level of detail decomposition of large data sets are wavelets. There are numerous examples on how to use wavelets to perform an efficient LoD based visualization of large data sets, yet those primarily focus on regular rectilinear grids [15,16]. For irregular grids, such as ICON, this becomes a bit more difficult, but it is, nevertheless, still possible. Here, so called icosahedral maps can be employed that are designed to fit the geometry of different cell configurations within the ICON model, as we have discussed in our prior publication [17]. As this research is still work in progress, this section summarizes our previous accomplishments, and outlines our current efforts in this direction.

Icosahedral maps contain the connectivity information in ICON in a highly structured two-dimensional hexagonal representation and facilitate the execution of a multi-resolution analysis on ICON data by applying a hexagonal version of the discrete wavelet transform. A global ICON grid is thereby broken down into ten diamonds, in which the data can be accessed and processed more easily [17], see also Fig. 6:

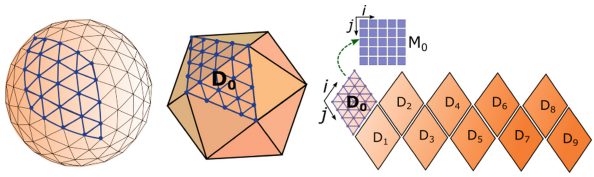

Fig. 6. Global ICON grid unfolded into a net consisting of ten diamonds. The vertex information of each diamond is stored in a 2D rectangular grid that corresponds to the hexagonal lattice associated with that diamond.

Figure 5 shows the principle of this wavelet decomposition for all three ICON grids using a low resolution global ocean simulation. The left column shows the original data, followed by a mapping onto a triangular grid (icosahedral maps), a coarsened grid (lower level of detail) and two reconstructions. In order to observe how the data responds to compression, a quantile thresholding was applied, keeping only those detail coefficients whose magnitudes falls within a specified percentile range. The last two columns in Fig. 5 shows that the wavelet responds very well to compression, i.e. the discarding of certain details. The reconstruction for two different thresholds – 10% and 95% – is shown, and displays that even a very aggressive compression of 95% – i.e. only 5% of the details are retained – is feasible. While this is still only work in progress, it clearly shows that a wavelet based decomposition and lossy compression of ICON data is possible and desirable. The here described wavelet based decomposition of the data would be performed in-situ, and will be implemented on the C++ side of the Catalyst adaptor in the form of an additional data flow branch, as is already outlined in Fig. 2. Researchers at NCAR in Boulder, Colorado, have looked into possible gains and losses by using various lossy compression algorithms, and shown that compression ratios of 1:5 are feasible, without even impairing the statistical signal of the data [18,19].

VAPOR, an interactive 3D visualization platform that is also developed at NCAR can be used to load and display such wavelet decomposed data sets. In fact, VAPOR features the so called VAPOR Data Collection (VDC) data model that allows users to progressively load and visualize their data, thus allowing an interactive visualization of terascale data sets on commodity hardware [15,20]. Initially designed to handle regular rectilinear grids only, it was more recently

extended to additionally support the UGRID netCDF-CF[7] standard, i.e. allowing one to also load and display model data on irregular grids, such as MPAS and ICON simulation output.

4 Summary and Conclusion

We have discussed the current status of in-situ visualization/processing at DKRZ, along with a few other ideas to handle some of the extremely large data sets that are produced by current climate simulations. The work presented is still work in progress, and although currently none of the work discussed is used in production mode, it is expected that the in-situ visualization and processing techniques will transition within a few weeks once workflows that are easy to setup and deploy by climate scientists have been devised. After that, other in-situ feature detection and tracking, such as the discussed ocean eddy census, will be added. Steering, as is developed as part of the exascale visualization initiative SENSEI, is also interesting, but probably a topic that lies – at least for us – a bit further into the future [9]. The wavelet decomposition and compression that was outlined in the previous Sect. 3 is implemented as standalone prototype, and needs to be relocated and further optimized into the Catalyst adaptor.

Other interesting aspects include the in-situ generation of high-quality visualizations using raytracing. As our current cluster is based on Intel CPUs, with relatively outdated GPUs, our current choice is OSPRay. Nevertheless, both OSPRay and OptiX are able to create superior quality renderings and are already used at DKRZ within the conventional post visualization workflow [21,22]. A transition of raytracing to in-situ will probably also take some more time.

Furthermore, just a visual display and analysis of large simulations may not be very useful in the near future, as the data is so massive that a single user will have problems finding and detecting the interesting features in the visual output. Here the currently popular machine learning might prove useful to automatically check the plausibility of a simulation, and also to identify outliers and extreme weather/climate events and direct the attention of the researcher directly onto those regions/time steps.

References

1. Zängl, G., Reinert, D., Ripodas, P., Baldauf, M.: The ICON (ICOsahedral non-hydrostatic) modelling framework of DWD and MPI-M: description of the non-hydrostatic dynamical core. Q. J. Roy. Meteorol. Soc. **141**(687), 563–579 (2015)
2. Heinze, R., et al.: Large-eddy simulations over Germany using ICON: a comprehensive evaluation. Q. J. Roy. Meteorol. Soc. **143**, 69–100 (2016)
3. Wes Bethel, E., Childs, H., Hansen, C., et al.: High Performance Visualization: Enabling Extreme-Scale Scientific Insight. Chapman and Hall/CRC Press, London (2016). ISBN 9781138199613

[7] http://ugrid-conventions.github.io/ugrid-conventions/.

4. Bauer, A., et al.: In situ methods, infrastructures, and applications on HPC platforms -a state-of-the-art (STAR) report. In: Computer Graphics Forum, vol. 35, no. 3 (2016)
5. Childs, H., et al.: VisIt: an end-user tool for visualizing and analyzing very large data. In: High Performance Visualization-Enabling Extreme-Scale Scientific Insight, pp. 357–372 (2012)
6. Ahrens, J., Geveci, B., Law, C.: ParaView: An End-User Tool for Large Data Visualization. Visualization Handbook. Elsevier (2005). ISBN 13: 978-0123875822
7. Ayachit, U.: The ParaView Guide: A Parallel Visualization Application. Kitware (2015). ISBN 978-1930934306
8. Larsen, M., et al.: The ALPINE in situ infrastructure: ascending from the ashes of Strawman. In: Proceedings of the in Situ Infrastructures on Enabling Extreme-Scale Analysis and Visualization (ISAV 2017), pp. 42–46. ACM, New York (2017). https://doi.org/10.1145/3144769.3144778
9. Ayachit, U., et al.: The SENSEI generic in situ interface. In: Second Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization (ISAV), pp. 40–44 (2016)
10. Ayachit, U., et al.: ParaView catalyst: enabling in situ data analysis and visualization. In: Proceedings of the First Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization (ISAV 2015), pp. 25–29. ACM (2015)
11. Woodring, J., Petersen, M., Schmeiss, A., Patchett, J., Ahrens, J., Hagen, H.: In situ eddy analysis in a high-resolution ocean climate model. IEEE Trans. Vis. Comput. Graph. **22**(1), 857–866 (2016)
12. Ahrens, J., Jourdain, S., O'Leary, P., Patchett, J., Rogers, D.H., Petersen, M.: An image-based approach to extreme scale in situ visualization and analysis. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 424–434. IEEE Press (2014)
13. Banesh, D., Schoonover, J.A., Ahrens, J.P., Hamann, B.: Extracting, visualizing and tracking mesoscale ocean eddies in two-dimensional image sequences using contours and moments. In: Workshop on Visualisation in Environmental Sciences (EnvirVis) (2017)
14. Schiffer, R.A., Rossow, W.B.: The international satellite cloud climatology project (ISCCP): the first project of the world climate research programme. Bull. Amer. Meteorol. Soc. **64**, 779–784 (1983)
15. Clyne, J., Rast, M.: A prototype discovery environment for analyzing and visualizing terascale turbulent fluid flow simulations. In: Proceedings of Visualization and Data Analysis, pp. 284–294 (2005)
16. Balsa Rodriguez, M., et al.: State-of-the-art in compressed GPU-based direct volume rendering. Comput. Graph. Forum **33**, 77–100 (2014)
17. Jubair, M.I., Alim, U., Röber, N., Clyne, J., Mahdavi-Amiri, A.: Icosahedral maps for a multiresolution representation of earth data. VMV: Vision Modeling and Visualization, Bayreuth, Germany (2016)
18. Baker, A.H., Xu, H., Hammerling, D.M., Li, S., Clyne, J.P.: Toward a multimethod approach: lossy data compression for climate simulation data. In: Kunkel, J.M., Yokota, R., Taufer, M., Shalf, J. (eds.) ISC High Performance 2017. LNCS, vol. 10524, pp. 30–42. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67630-2_3
19. Baker, A.H., Hammerling, D.M., Mickelson, S.A., et al.: Evaluating lossy data compression on climate simulation data within a large ensemble. Geoscientific Model Dev. **9**, 4381–4403 (2016)

20. 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 J. Phys. **9**, 301 (2007)
21. Wald, I., et al.: OSPRay - a CPU ray tracing framework for scientific visualization. IEEE Trans. Vis. Comput. Graph. **23**(1), 931–940 (2017)
22. Parker, S.G., et al.: OptiX: a general purpose ray tracing engine. ACM Trans. Graph. **29**, 66 (2010)