Reducing Occlusion in Cinema Databases through Feature-Centric Visualizations

Roxana Bujack, David H. Rogers and James Ahrens

Data Science at Scale team, Los Alamos National Laboratory, Los Alamos, USA
E-mail: bujack@lanl.gov, dhr@lanl.gov, ahrens@lanl.gov

***Abstract*—In modern supercomputer architectures, the I/O capabilities do not keep up with the computational speed. Image-based techniques are one very promising approach to a scalable output format for visual analysis, in which a reduced output that corresponds to the visible state of the simulation is rendered in-situ and stored to disk. These techniques can support interactive exploration of the data through image compositing and other methods, but automatic methods of highlighting data and reducing clutter can make these methods more effective. In this paper, we suggest a method of assisted exploration through the combination of feature-centric analysis with image space techniques and show how the reduction of the data to features of interest reduces occlusion in the output for a set of example applications.**

***Index Terms*—occlusion, feature, in situ, image space, Cinema, pattern detection, moment invariants**

I. Introduction

The development of modern supercomputer architectures tends to increase the computational capacity faster than the I/O bandwidth [22]. Because of this, the visualization community has developed in-situ analysis techniques that can identify areas of interest in data and reduce the amount of data written to disk. These techniques can help scientists manage the output of their simulations in order to make large results more manageable and explorable.

For visual inspection of large data, image-based approaches such as Cinema [4] provide a compact, interactive representation of data that is extremely scalable. Cinema is an analysis ecosystem, made up of a set of database specifications, writers, algorithms, and viewers 1. It is an innovative way of capturing, storing, and exploring extreme scale scientific data. It is a highly interactive image-based approach to data analysis and exploration that promotes investigation of large scientific datasets, and is easily integrated into existing workflows through extensions to widely used open source tools. This novel approach supports interactive exploration of a wide range of results, while still significantly reducing data movement and storage.

In exascale applications, we expect to compute 10^{18} flops per second on data of peta- to exabytes. Regardless of the size of the data, the size of an image will remain in the area of 10^6 pixels. Cinema provides writers and readers that capture a dataset from different angles, and viewers that can later display them coherently providing the illusion of interacting with the data itself. Cinema viewers can composite images, which allows a wide variety of interactions, including turning elements 'on' and 'off', and interactive slice, color, and opacity controls, making the system very flexible for exploration. However, assisted exploration, in which various features are highlighted by in-situ algorithms, can vastly improve a scientist's exploration of the data by suggesting areas of interest for interactive exploration.

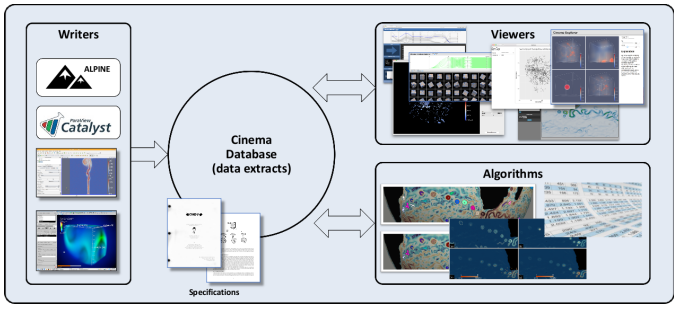


Fig. 1: Diagram of the Cinema ecosystem showing how writers, database specifications, viewers and algorithms work together to capture, analyze, and visualize data artifacts from a variety of scientific use cases. A common element of a Cinema database is a set of images that provide an interactive dataset for exploring even extremely large datasets.

Image space techniques work well if the majority of the data lies on surfaces, Figure 2, but for general three-dimensional data, they suffer from occlusion, like any 3D visualization, as we have only a two-dimensional space to display information. For each pixel, there is only one color that can be displayed, even though it has to represent all the data that lies on the line spanned by it and the observer's eye. That means that for everything we draw, we have to discard a myriad of information.

For this paper, we assume that the application scientists already have prior knowledge about the dataset and some feature in which they are particularly interested. We search for this pattern and use the result to prioritize the visualization and reduce the occlusion by rendering transparent all areas that do not resemble the feature. This allows the application scientist to analyze the areas of interest from all sides without being obscured by clutter.

We have released the open-source VTK module MomentIn-

LEVIA'18: Leipzig Symposium on Visualization in Applications 2018
This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

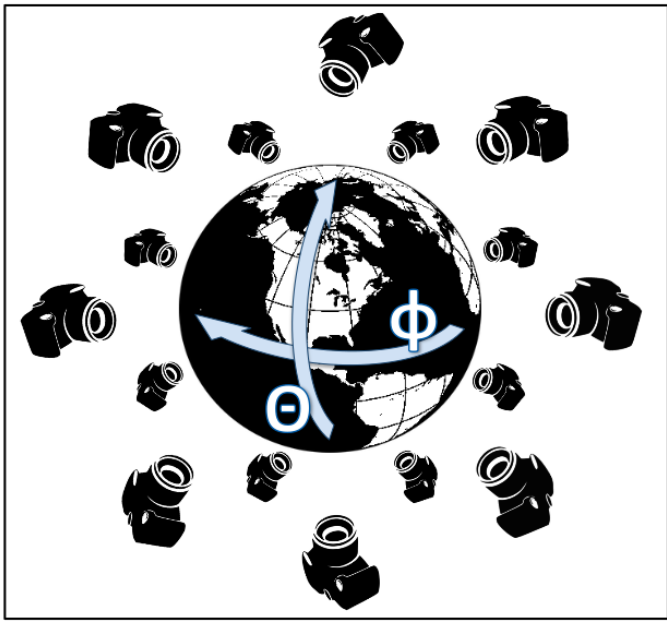


Fig. 2: Generating a Cinema database through snapshots of the data from different angles. The composition of images allows an interactive exploration of the data. However, the inside of the globe can not be seen.

variants [2], [7], [45]. Based on the algorithm developed by Bujack et al. [18], this module performs rotation invariant pattern detection on two- and three-dimensional scalar, vector, and matrix fields. The architecture with example images can be found in Figure 3. Its major advantage is that it is rotation invariant without requiring known point to point relations. This means that we provide only one instance of the template pattern and the technique will find any arbitrarily oriented match.

The combination of this black-box pattern detection algorithm with Cinema creates visualizations that meet the three requirements of in-situ visualization stated by Ahrens et al [4]:

- To preserve important elements of the simulations.
- To significantly reduce the data needed to preserve these elements.
- To offer as much flexibility as possible for post-processing exploration.

The main contributions of this paper are:

- Release of an open-source implementation of the moment-based rotation-invariant pattern detection algorithm [18] through VTK.
- Application in a workflow to reduce occlusion in 3D in-situ visualizations, like Cinema databases, and
- Demonstration of the concept for direct and for sparse visualization techniques in two example applications.

II. Related Work

In situ visualization and analysis is an important mechanism for extreme scale simulations because it can impact the cost of data movement and storage for traditional post processing methods [12], [37]. Since we want to explore the data visually, image based techniques [47] are a straight forward strategy [21], [27], [51].

We will make use of Cinema [4], [53], which provides tremendous data reduction while preserving a maximum of flexibility in the exploration. It has been used in many applications such as simulations of the oceans [10], [11], [14], [40], cosmology [46], or the mixing of fluids of different concentrations of solvents [36].

In general interactive visualization environments, occlusion can be reduced through filtering, cutting away parts of the data, or adjusting transparency through lens tools on the fly [15], [29]. Since we plan on generating visualizations in-situ, there is no easy way of interacting with the data like that. Instead, we rely on methods that can automatically detect features of interest, and determine what can be removed from the scene.

Feature detection is a method in image processing that locates areas of interest in an image [50]. In scalar fields, these may be edges, corners, or closed regions [33], [35], [52] and in vector fields vortices or topological structures [43], [44]. While many feature detection algorithms are very specialized, for example to vector field topology, spatial regions, or certain values in the data [39], [49], we will focus on the detection of general patterns in this paper [6], [16]. Cross-correlation is a basic technique to determine the similarity between arbitrary images [17], [54]. In contrast to SIFT [38] or comparable techniques [28], it does not require point-to point relations, which are difficult to come by in smooth data. The problem is that usually the orientation of the pattern is unknown and to make sure that it is captured by the cross-correlation, we would have to look for all possible rotations of the pattern. This can be avoided through rotation-invariant pattern detectors such as moment invariants [25].

Moment Invariants were introduced to image processing by Hu [26]. Dirilten and Newman showed that the moment tensor contractions to zeroth order are invariant under orthogonal transformations [24] and Pinjo, Cyganski, and Orr [42] calculated 3D orientation estimation from moment contraction to first order moments. Suk and Flusser proposed to calculate the complete set of tensor contractions and remove redundancy by skipping the linearly dependent ones in [48]. Langbein and Hagen [30], [31] showed that the tensor contraction method can be generalized to arbitrary tensor fields.

For our implementation, we use the powerful moment invariants algorithm by Bujack et al. [18], which can detect patterns independent of their orientation in two-dimensional as well as three-dimensional scalar-, vector-, and tensor-valued data.

III. Workflow

We provide an open-source implementation of the rotation-invariant pattern detection algorithm in the VTK [7], [45] module MomentInvariants. The source code and detailed description of its functionality can be found in [2]. This module

detects patterns in 2D and 3D scalar-, vector-, and matrix-valued datasets.

The module takes the dataset and the pattern as input and it performs the pattern detection independent of how the pattern and the match are aligned w.r.t. each other. The filter vtkSimilarityBalls outputs the similarity value stored in a ball of the radius equal to the detected match's scale around all local maxima of similarity in a vtkImageData format. If a neighboring location or a neighboring scale shows a better match to the pattern, the entry in the output dataset is zero. A symbolic scheme of this workflow is shown in Figure 3.

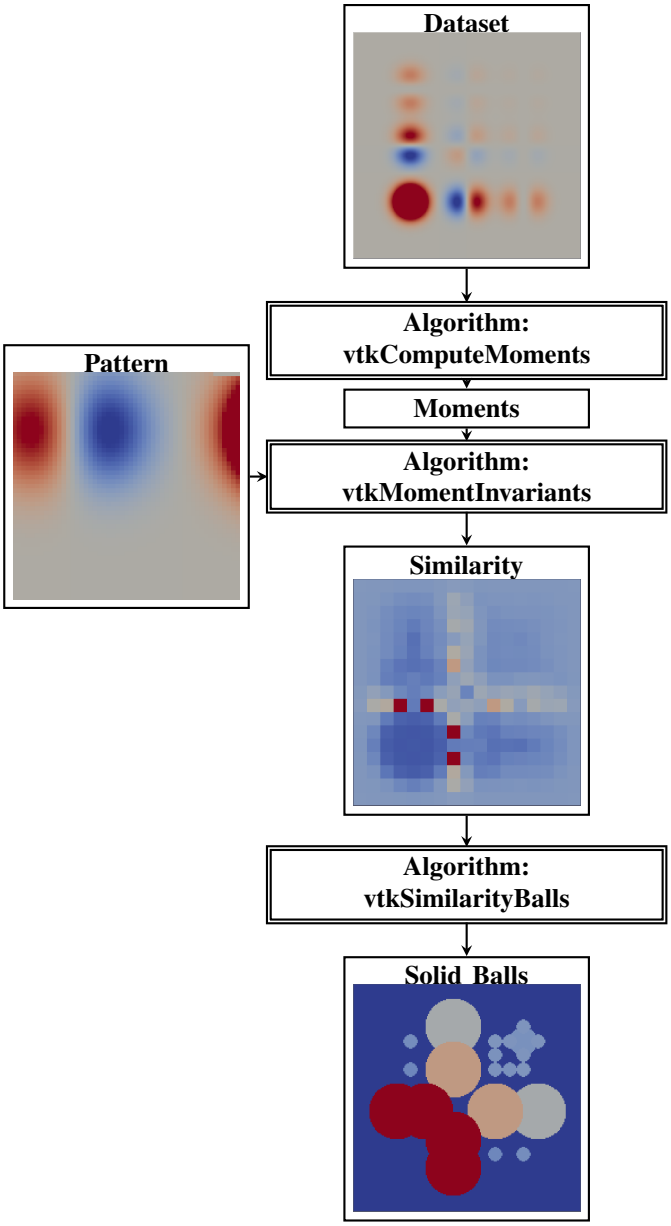


Fig. 3: Generation of the spatial areas of interest through the VTK module MomentInvariants with example images.

The filter vtkSimilarityBalls in this module produces fields that are explicitly tailored towards visualizing the similarity. It can put out the surfaces of spheres/circles, which are good for 2D visualizations [19] because they can be laid over a visualization of the field, or solid spheres, which are useful for 3D visualizations [20]. The information of the match is encoded in the spheres as follows:

- The center of the sphere corresponds to the location of the match.
- The diameter of the sphere encodes the size at which the match occurred.
- The scalar value of the sphere is the similarity to the pattern, i.e., the reciprocal of the Euclidean distance of the normalized moments.

In this paper, we concentrate on the reduction of occlusion in 3D data. Here, the extended similarity information can be used in two major ways to reduce occlusion in 3D data. On one hand, for visualization techniques that place discrete objects, e.g. glyphs or streamlines, we can use a seeding strategy that correlates with the similarity. Alternately, for direct visualization techniques, e.g. volume rendering or nested isosurfaces, we can render parts of the scene transparent in correlation with the similarity, resembling [49].

We will provide an example use case for each approach. In our examples, we will use ParaView [3], [8] for the execution of the workflow, because

- It is based on vtk and can import its filters.
- It provides most basic visualization algorithms.
- It can be connected to a simulation through its in-situ library Catalyst [9], [13].
- It comes with a built-in capability of directly exporting and importing Cinema databases.

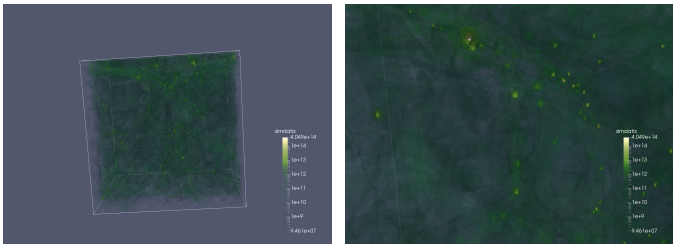
IV. Direct Visualizations

In this section, we show how to reduce occlusion for direct visualization techniques, by which we refer to methods that are applied densely to the overall domain in contrast to indirect visualization techniques that are places at discrete seed points. We use a real world visualization scenario in which pattern detection is applied to a visualization of the NGC [23] ECP application Nyx [5] with stacked, semi-transparent isosurfaces and one with volume rendering.

The simulation models gravity-driven structure formation in the universe and particularly targets accurate reconstruction of the gas in the intergalactic medium. It is intended to study the large-scale cosmological dark matter activities. The NYX flagship simulation is the world's biggest hydrodynamical cosmological simulation and among the largest scientific simulations in general. For this use case, we use a single time step towards the end of the simulation when clusters of mass have already formed and a much lower resolution of only 128^3 points. The scalar variable visualized is the density.

Three-dimensional visualizations generally suffer from occlusion, which is exacerbated by direct visualization techniques. There are techniques that allow for more than the surface of the object at hand to contribute, for example volume rendering or the use of partial transparency, but they do not

solve the problem. Figure 4 shows how difficult it is to interpret the dataset, even though the stacked isosurfaces and the colormap have been chosen very carefully to adequately represent the underlying value range through a logarithmic transformation.



(a) Overview. (b) Close-up view.

Fig. 4: Ten semi-transparent isosurfaces of the Nyx data equidistantly spaced and color mapped in a logarithmic scale.

As an example feature of interest, we chose the combination of two halos, i.e., areas of high density that indicate the existence of galaxies, that have the same size and are about as far away from each other as their diameter, Figure 6a. The pattern detection algorithm finds regions that are similar to the pattern, Figure 6b. We use this information to reduce the occlusion by rendering fully transparent all areas of the isosurfaces that do not resemble the feature, Figure 7. Generating the Cinema database from this reduced view allows the application scientist to analyze the areas of interest from all sides without being obscured by occlusion from uninteresting areas.

In ParaView, we compute isosurfaces using the Contour filter. Its output and the similarity with the solid balls are both input into the Programmable filter. For each point on the isosurfaces, it assigns values for RGB plus opacity. The RGB values are chosen to copy the same colormap that we used for the original semi-transparent isosurfaces that visualize the whole dataset from Figure 4. The most important part is the adjustment of the opacity. For this project, we chose to set everything outside the areas of interest perfectly transparent, but it would be trivial to assign a low opacity if the application scientist would want to highlight the areas of interest while still providing context. For the visible parts, we decided that we wanted to show every match with a similarity above a threshold of the pattern no matter at which scalar range of the density variable it occurs, but we also wanted to emphasize the higher value ranges so that in the case of nested contours the ones deeper inside have a chance to be perceived through a higher opacity. We chose an opacity of zero if the similarity is below 4, and otherwise $0.1 + 0.9 * \rho^2$, where ρ corresponds to the logarithm of the density scaled to lie in $[0, 1]$. An overview of this part of the workflow can be found in Figure 5.

Analogous to the stacked isosurfaces, we can apply this workflow to volume rendering. In Figure 8, we used the same colormap for both visualizations. But for the feature-centric visualization, we set the opacity to zero if the area does not resemble the pattern. This can be interpreted as the transfer

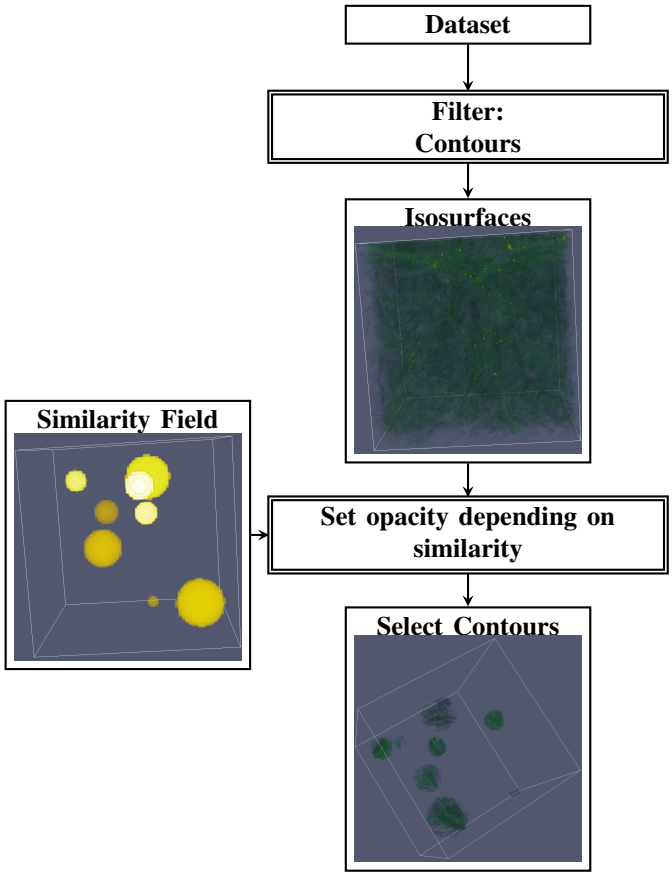
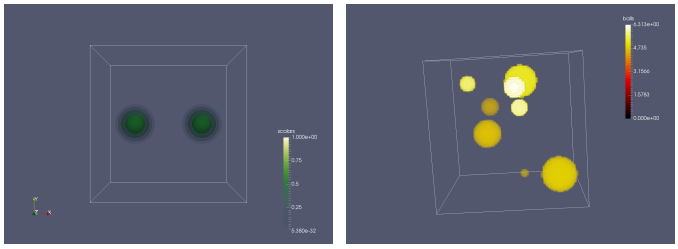


Fig. 5: Generation of the decluttered visualization of the nested isosurfaces rendered only in areas of high similarity to the pattern.

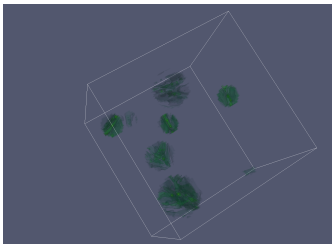


(a) Isosurface visualization of the pattern. (b) Volume rendering of the similarity maxima.

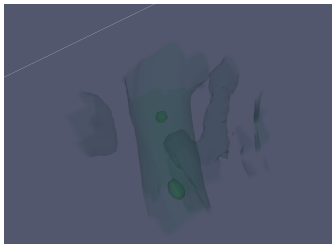
Fig. 6: The pattern and the dataset's similarity to it.

function to be a function in two variables, the value of the simulation and the similarity.

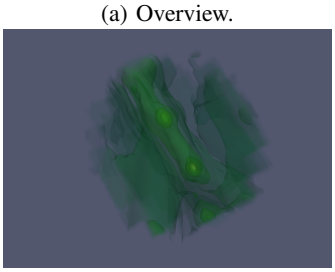
The final visualization with reduced clutter was exported to a Cinema database directly through ParaView's Export Scene capabilty. The combined databases showing the isosurfaces of the pattern and the dataset together with the a volume rendering of the similarity output and the final visualization can be explored with the new browser-based Cinema spec-D viewer by opening selectContours.cdb/index.html from the supplementary material in Firefox.



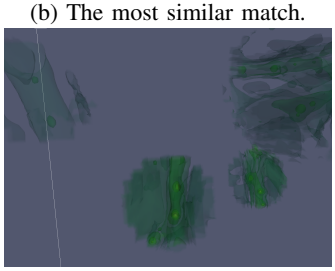
(a) Overview.



(b) The most similar match.

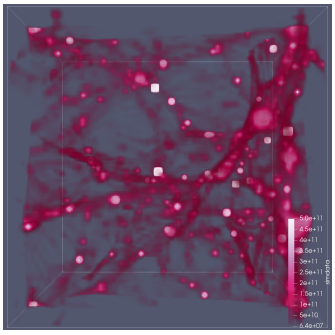


(c) Another good match

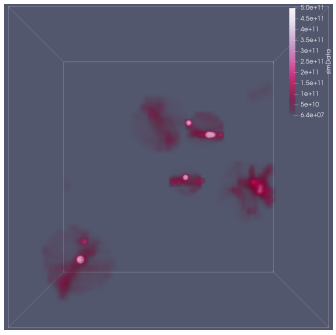


(d) Closeup into several matches.

Fig. 7: The ten semi-transparent isosurfaces of the Nyx data from Figure 4 after rendering transparent everything that is not of interest.



(a) Volume rendering of the whole dataset.



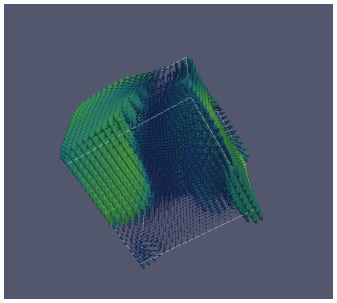
(b) Volume rendering reduced to the regions of interest.

Fig. 8: Application of the workflow to volume rendering.

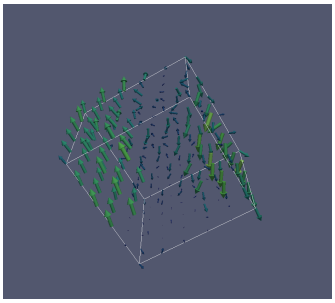
V. Indirect Visualizations

Indirect visualization elements, like glyphs, stream- or pathlines can be treated like the direct ones from the previous section. But since, they come with a location where they are placed, we can also go down a different route and select the seeding locations based on the similarity. In this section, we show how this can be done at one example of arrow glyphs and one of streamlines applied to a turbulent vector field.

The homogeneous buoyancy driven turbulence dataset [1], [34] produced by LANL's Direct Numerical Simulation (DNS) code, available atJohn's Hopkins Turbulence Database [32], [41] are the result of solving an incompressible two-fluid Navier-Stokes equation. It covers a triply periodic domain with up to $1,024^3$ points. For the simulation, two fluids are initialized randomly. The flow starts from rest. The fluids start moving in opposite direction due to gravity and differential buoyancy forces. The scientists use the simulation to study the increase and decay of turbulence fluctuations as the fluids become molecularly mixed. Each of the 1015 time frames contains the density, 3 velocity components, and pressure at the grid points. In this example, we use the 200-th time step of the of the simulation with a resolution of 128^3. We visualize the three-dimensional velocity field using arrow glyphs. Figure 9 illustrates the problem of understanding the structure of the data from uniformly seeded glyphs in the dataset. The velocity magnitude is encoded in the size and color of the arrows and the colormap. The direction of the flow is given by the direction of the arrow.



(a) Placing too many arrow glyphs results in occlusion.



(b) Placing too few arrow glyphs misses the internal structure.

Fig. 9: The optimal placement of glyphs is not a Goldy-Locks-problem. There is no right amount in the middle.

As in the previous example, we use the MomentInvariants module of VTK to generate a feature-centric visualization that only displays areas that resemble a pattern of interest, for which we chose the saddle sufficing $e^{-\frac{1}{2}(x^2+y^2)}(y,x,0)^T$ shown in Figure 11a. We place the arrow glyphs proportional to the similarity information of the solid balls encoding the extent of the detected pattern. In particular, we generate a random number for each point of the original dataset. If its value is smaller than the similarity appropriately scaled, it will be used as a seed point for a glyph in the final visualization. An illustration of the workflow can be found in Figure 10 and the detected features in Figure 11.

Alternatively, we can use the selected points to seed other indirect visualization elements, like streamlines, Figure 12.

It is particularly interesting how there is a whole axis along which the dataset shows the saddle like behavior of the pattern, Figures 11, 12. It would have been difficult to find that using 2D slice views through the dataset because this it is not aligned with any of the coordinate axes.

The combined databases showing streamlines of the pattern and the dataset with different seeding strategies can be explored with the Cinema spec-D viewer in a browser. A screenshot shows a dense uniform and a sparse uniform seeding in the top of Figure 13. The bottom displays the pattern and the feature-centric seeding.

VI. Discussion

We have released an open source implementation of the rotation-invariant pattern detection algorithm presented in [18]

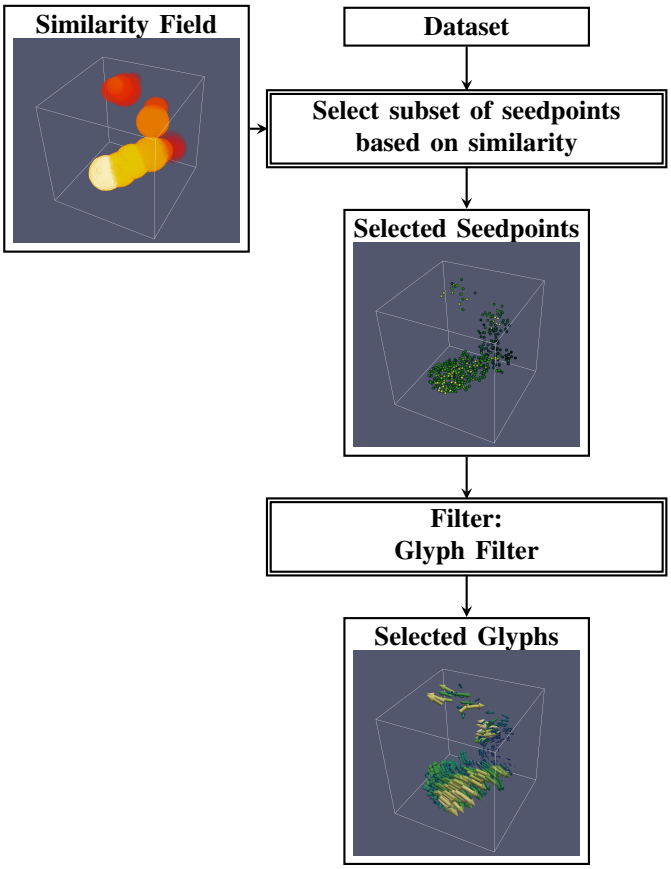
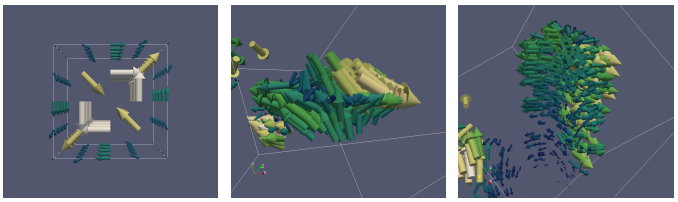


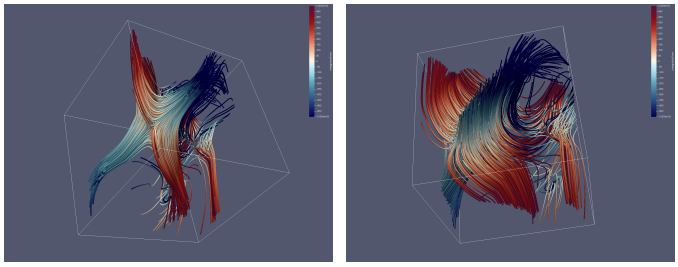
Fig. 10: Generation of the decluttered visualization of arrow glyphs rendered only in areas of high similarity to the pattern.



(a) The pattern: a saddle. (b) The front of the saddle axis. (c) The side of the saddle axis.

Fig. 11: The main occurance of the saddle in the turbulence dataset is not isolated, but extended along an axis.

and proposed a workflow for how it can be used to reduce occlusion in in-situ visualizations, such as for the generation of Cinema [4] databases. Further, we demonstrated its effect in two example applications. First, we used it to highlight areas of interest in the NGC [23] application Nyx [5] with stacked, semi-transparent isosurfaces and volume rendering as two representatives of direct visualization methods. Second, we applied it to steer the seeding of arrow glyphs and streamlines in the homogeneous buoyancy driven turbulence dataset [1], [34] as two representative examples for indirect visualization methods.



(a) The front of the saddle axis. (b) The side of the saddle axis.

Fig. 12: Visualization of the saddle axis in the turbulence data with stream tubes. The color represents the integration time, red forward, blue backward.

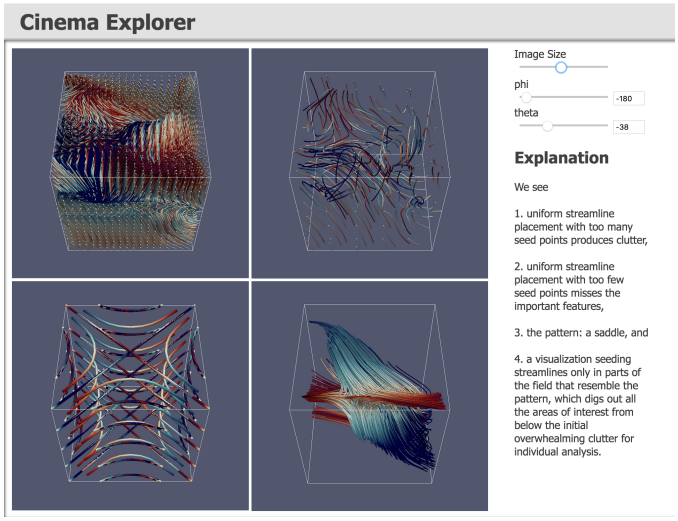


Fig. 13: Screenshot of the browser-based Cinema spec-D viewer showing different streamline seeding strategies for the turbulence data in a combined view.

In both cases, the reduction of the displayed elements to regions of interest reduced clutter and allowed interactive exploration from all sides in the post hoc analysis with Cinema even though the original data was not constrained to surfaces.

We hope to apply this workflow in closer collaboration with the domain scientists and integrate it into their workflow in future work. Learning which features are of particular interest to them might help us identify requirements for feature detection methods that do not require a priori information.

Acknowledgment

This work is published under LA-UR-18-25355. It was supported by the National Nuclear Security Administration (NNSA) Advanced Simulation and Computing (ASC) Program.

References

[1] Johns hopkins turbulence databases, homogeneous buoyancy driven turbulence data set. http://turbulence.pha.jhu.edu/Homogeneous_buoyancy_driven_turbulence.aspx. Accessed: 2018-06-14.

[2] Vtk module moment invariants. https://gitlab.kitware.com/vtk/MomentInvariants. Accessed: 2018-06-14.

[3] J. Ahrens, B. Geveci, and C. Law. Paraview: An end-user tool for large-data visualization. *The Visualization Handbook*, page 717, 2005. LA-UR-03-1560.

[4] J. Ahrens, S. Jourdain, P. O'Leary, J. Patchett, D. H. Rogers, and M. Petersen. 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*, pages 424–434. IEEE Press, 2014.

[5] A. S. Almgren, J. B. Bell, M. J. Lijewski, Z. Luki?, and E. V. Andel. Nyx: A massively parallel amr code for computational cosmology. *The Astrophysical Journal*, 765(1):39, 2013.

[6] A. Amir. Multidimensional Pattern Matching: A Survey. Technical report, 1992.

[7] L. S. Avila, S. Barre, R. Blue, B. Geveci, A. Henderson, W. A. Hoffman, B. King, C. C. Law, K. M. Martin, and W. J. Schroeder. *The VTK User's Guide*. Kitware New York, 2010.

[8] U. Ayachit. *The ParaView Guide: A Parallel Visualization Application*. Kitware, Inc., USA, 2015.

[9] U. Ayachit, A. Bauer, B. Geveci, P. O'Leary, K. Moreland, N. Fabian, and J. Mauldin. 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*, pages 25–29. ACM, 2015.

[10] D. Banesh, J. A. Schoonover, J. P. Ahrens, and B. Hamann. Extracting, Visualizing and Tracking Mesoscale Ocean Eddies in Two-dimensional Image Sequences Using Contours and Moments. In K. Rink, A. Middel, D. Zeckzer, and R. Bujack, editors, *Workshop on Visualisation in Environmental Sciences (EnvirVis)*. The Eurographics Association, 2017.

[11] D. Banesh, J. Wendelberger, M. Petersen, J. Ahrens, and B. Hamann. Change Point Detection for Ocean Eddy Analysis. In K. Rink, D. Zeckzer, R. Bujack, and S. Jnicke, editors, *Workshop on Visualisation in Environmental Sciences (EnvirVis)*. The Eurographics Association, 2018.

[12] A. C. Bauer, H. Abbasi, J. Ahrens, H. Childs, B. Geveci, S. Klasky, K. Moreland, P. O'Leary, V. Vishwanath, B. Whitlock, et al. In situ methods, infrastructures, and applications on high performance computing platforms. In *Computer Graphics Forum*, volume 35, pages 577–597. Wiley Online Library, 2016.

[13] A. C. Bauer, B. Geveci, and W. Schroeder. The paraview catalyst user?s guide, 2013.

[14] A. S. Berres, T. L. Turton, M. Petersen, D. H. Rogers, and J. P. Ahrens. Video compression for ocean simulation image databases. In *Workshop on Visualisation in Environmental Sciences (EnvirVis). The Eurographics Association*, 2017.

[15] E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. D. DeRose. Toolglass and magic lenses: the see-through interface. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 73–80. ACM, 1993.

[16] C. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2006.

[17] L. G. Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24:325–376, 1992.

[18] R. Bujack and H. Hagen. Moment Invariants for Multi-Dimensional Data. In E. Ozerslan, T. Schultz, and I. Hotz, editors, *Modelling, Analysis, and Visualization of Anisotropy*, Mathematica and Visualization. Springer Basel AG, 2017.

[19] R. Bujack, I. Hotz, G. Scheuermann, and E. Hitzer. Moment Invariants for 2D Flow Fields via Normalization in Detail. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 21(8):916–929, Aug 2015.

[20] R. Bujack, J. Kasten, I. Hotz, G. Scheuermann, and E. Hitzer. Moment Invariants for 3D Flow Fields via Normalization. In *IEEE Pacific Visualization Symposium, PacificVis 2015 in Hangzhou, China*, 2015.

[21] J. Chen, I. Yoon, and W. Bethel. Interactive, internet delivery of visualization via structured prerendered multiresolution imagery. *IEEE Transactions on Visualization and Computer Graphics*, 14(2):302–312, 2008.

[22] H. Childs, D. Pugmire, S. Ahern, B. Whitlock, M. Howison, Prabhat, G. Weber, and E. W. Bethel. Extreme Scaling of Production Visualization Software on Diverse Architectures. *IEEE Computer Graphics and Applications (CG&A)*, 30(3):22–31, May/June 2010.

[23] D. J. Daniel and A. L. Hungerford. Lanl asc advanced technology development and mitigation: Next-generation code project (ngc). Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2016.

[24] H. Dirilten and T. G. Newman. Pattern matching under affine transformations. *Computers, IEEE Transactions on*, 100(3):314–317, 1977.

[25] J. Flusser, B. Zitova, and T. Suk. *2D and 3D Image Analysis by Moments*. John Wiley & Sons, 2016.

[26] M.-K. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187, 1962.

[27] A. Kageyama and T. Yamada. An approach to exascale visualization: Interactive viewing of in-situ visualization. *Computer Physics Communications*, 185(1):79–85, 2014.

[28] E. Karami, S. Prasad, and M. Shehata. Image matching using sift, surf, brief and orb: Performance comparison for distorted images. *arXiv preprint arXiv:1710.02726*, 2017.

[29] D. A. Keim. Information visualization and visual data mining. *IEEE transactions on Visualization and Computer Graphics*, 8(1):1–8, 2002.

[30] M. Langbein. *Higher Order Moment Invariants and their Applications*. PhD Dissertation, Technical University Kaiserslautern, 2014.

[31] M. Langbein and H. Hagen. A Generalization of Moment Invariants on 2D Vector Fields to Tensor Fields of Arbitrary Order and Dimension. In G. Bebis, R. Boyle, B. Parvin, D. Koracin, Y. Kuno, J. Wang, R. Pajarola, P. Lindstrom, A. Hinkenjann, M. Encarnao, C. Silva, and D. Coming, editors, *Advances in Visual Computing*, volume 5876 of *Lecture Notes in Computer Science*, pages 1151–1160. Springer Berlin Heidelberg, 2009.

[32] Y. Li, E. Perlman, M. Wan, Y. Yang, C. Meneveau, R. Burns, S. Chen, A. Szalay, and G. Eyink. A public turbulence database cluster and applications to study lagrangian evolution of velocity increments in turbulence. *Journal of Turbulence*, (9):N31, 2008.

[33] T. Lindeberg. Edge Detection and Ridge Detection with Automatic Scale Selection. *Int. J. Comput. Vision*, 30(2):117–156, Nov. 1998.

[34] D. Livescu, C. Canada, K. Kanov, R. Burns, J. Pulido, et al. Homogeneous buoyancy driven turbulence data set. 2015.

[35] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[36] J. Lukasczyk, G. Aldrich, M. Steptoe, G. Favelier, C. Gueunet, J. Tierny, R. Maciejewski, B. Hamann, and H. Leitte. Viscous fingering: A topological visual analytic approach. In *Applied Mechanics and Materials*, volume 869, pages 9–19. Trans Tech Publ, 2017.

[37] K.-L. Ma. In situ visualization at extreme scale: Challenges and opportunities. *IEEE Computer Graphics and Applications*, 29(6):14–19, 2009.

[38] J.-M. Morel and G. Yu. Asift: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469, 2009.

[39] S. Muraki, I. Fujishiro, Y. Suzuki, and Y. Takeshima. Diffusion-based tractography: Visualizing dense white matter connectivity from 3d tensor fields. 2006.

[40] P. O'Leary, J. Ahrens, S. Jourdain, S. Wittenburg, D. H. Rogers, and M. Petersen. Cinema image-based in situ analysis and visualization of mpas-ocean simulations. *Parallel Computing*, 55:43–48, 2016.

[41] E. Perlman, R. Burns, Y. Li, and C. Meneveau. Data exploration of turbulence simulations using a database cluster. In *Proceedings of the 2007 ACM/IEEE conference on Supercomputing*, page 23. ACM, 2007.

[42] Z. Pinjo, D. Cyganski, and J. A. Orr. Determination of 3-D object orientation from projections. *Pattern Recognition Letters*, 3(5):351–356, 1985.

[43] A. Pobitzer, R. Peikert, R. Fuchs, B. Schindler, A. Kuhn, H. Theisel, K. Matkovic, and H. Hauser. The State of the Art in Topology-based Visualization of Unsteady Flow. *Computer Graphics Forum*, 30(6):1789–1811, September 2011.

[44] M. Roth. *Automatic Extraction of Vortex Core Lines and Other Line-Type Features for Scientific Visualization*. PhD Dissertation No. 13673, ETH Zurich, 2000. published by Hartung-Gorre Verlag, Konstanz, ISBN 3-89649-582-8.

[45] W. J. Schroeder, B. Lorensen, and K. Martin. *The visualization toolkit: an object-oriented approach to 3D graphics*. Kitware, 2004.

[46] C. Sewell, K. Heitmann, H. Finkel, G. Zagaris, S. T. Parete-Koon, P. K. Fasel, A. Pope, N. Frontiere, L.-t. Lo, B. Messer, et al. Large-scale compute-intensive analysis via a combined in-situ and co-scheduling workflow approach. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, page 50. ACM, 2015.

[47] H. Shum and S. B. Kang. Review of image-based rendering techniques. In *Visual Communications and Image Processing 2000*, volume 4067, pages 2–14. International Society for Optics and Photonics, 2000.

[48] T. Suk and J. Flusser. Tensor Method for Constructing 3D Moment Invariants. In *Computer Analysis of Images and Patterns*, volume 6855 of *Lecture Notes in Computer Science*, pages 212–219. Springer Berlin, Heidelberg, 2011.

[49] Y. Suzuki, I. Fujishiro, L. Chen, and H. Nakamura. Case study: hardware-accelerated selective lic volume rendering. In *Visualization, 2002. VIS 2002. IEEE*, pages 485–488. IEEE, 2002.

[50] A. Telea. *Data Visualization - Principles and Practice.* A K Peters, 2008.

[51] A. Tikhonova, C. D. Correa, and K.-L. Ma. Visualization by proxy: A novel framework for deferred interaction with volume data. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1551–1559, 2010.

[52] T. Tuytelaars and K. Mikolajczyk. Local Invariant Feature Detectors: A Survey. *Found. Trends. Comput. Graph. Vis.*, 3(3):177–280, July 2008.

[53] J. Woodring, J. P. Ahrens, J. Patchett, C. Tauxe, and D. H. Rogers. High-dimensional scientific data exploration via cinema. In *Data Systems for Interactive Analysis (DSIA), 2017 IEEE Workshop on*, pages 1–5. IEEE, 2017.

[54] B. Zitová and J. Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, 2003.