Applied Mechanics and Materials
ISSN: 1662-7482, Vol. 869, pp 9-19
doi:10.4028/www.scientific.net/AMM.869.9
© 2017 The Author(s). Published by Trans Tech Publications Ltd, Switzerland.
Submitted: 2017-02-13
Revised: 2017-05-04
Accepted: 2017-05-04
Online: 2017-08-30

Viscous Fingering: A Topological Visual Analytic Approach

Jonas Lukasczyk[1,a], Garrett Aldrich[2,3,b], Michael Steptoe[4,c], Guillaume Favelier[5,d], Charles Gueunet[5,e], Julien Tierny[5,f], Ross Maciejewski[4,g], Bernd Hamann[2,h], and Heike Leitte[1,i]

[1]Department of Computer Science, TU Kaiserslautern, Kaiserslautern 67663, Germany

[2]Department of Computer Science, University of California, Davis, CA 95616, U.S.A.

[3]CCS-7 Applied Computer Science, Los Alamos National Laboratory, Los Alamos, NM 87544, U.S.A.

[4]School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ 85287, U.S.A.

[5]Sorbonne Universites, UPMC Univ Paris 06, CNRS, LIP6 UMR 7606, France

[a]jl@jluk.de, [b]aldrich@lanl.gov, [c]msteptoe@mainex1.asu.edu, [d]guillaume.favelier@lip6.fr, [e]charles.gueunet@lip6.fr, [f]julien.tierny@lip6.fr, [g]rmacieje@asu.edu, [h]hamann@cs.ucdavis.edu, [i]leitte@cs.uni-kl.de

Keywords: Topological data analysis, data segmentation, feature tracking, ensemble data, viscous fingering, tracking graph.

Abstract. We present a methodology to analyze and visualize an ensemble of finite pointset method (FPM) simulations that model the viscous fingering process of salt solutions inside water. In course of the simulations the solutions form structures with increased salt concentration called viscous fingers. These structures are of primary interest to domain scientists as it is not clear when and where viscous fingers appear and how they evolve. To explore the aleatoric uncertainty embedded in the simulations we analyze an ensemble of simulation runs which differ due to stochastic effects. To detect and track the viscous fingers we derive a voxel volume for each simulation where fingers are identified as sub-volumes that satisfy geometrical and topological constraints. Properties and the evolution of fingers are illustrated through tracking graphs that visualize when fingers form, dissolve, merge, and split. We provide multiple linked views to compare, browse, and analyze the ensemble in real-time.

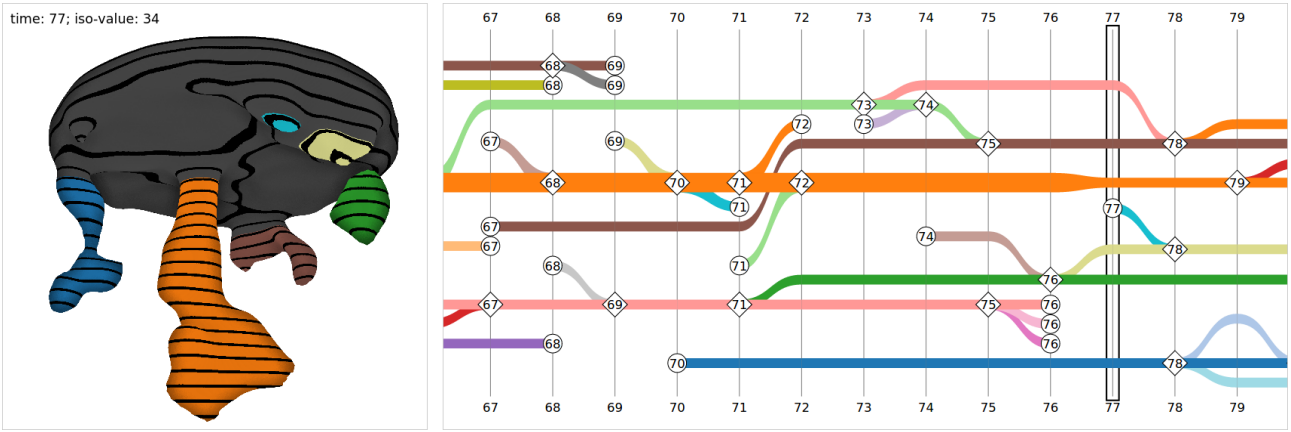


Fig. 1: Detail view of our visualization tool consisting of the 3D rendering of viscous fingers (left), and the tracking graph showing their evolution (right).

This article is an open access article under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0)

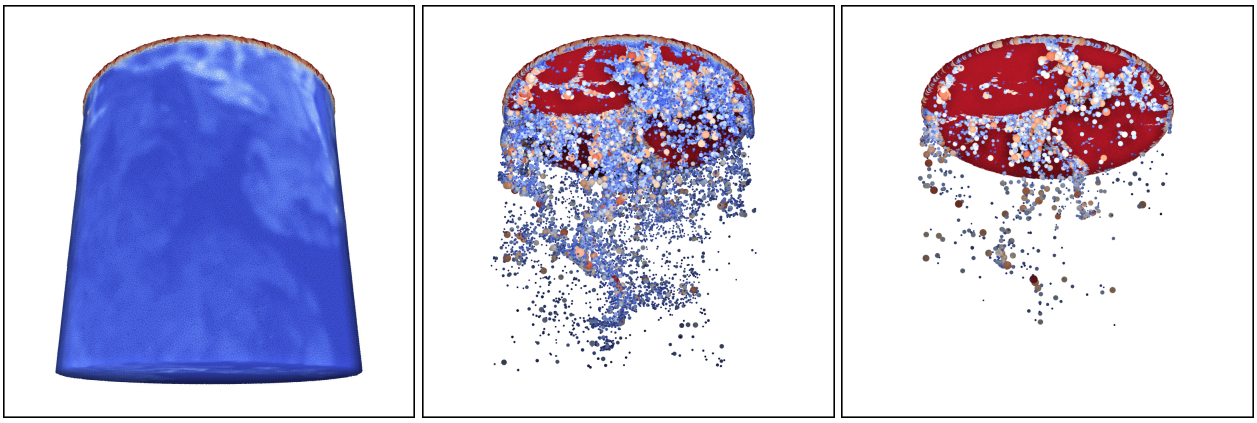


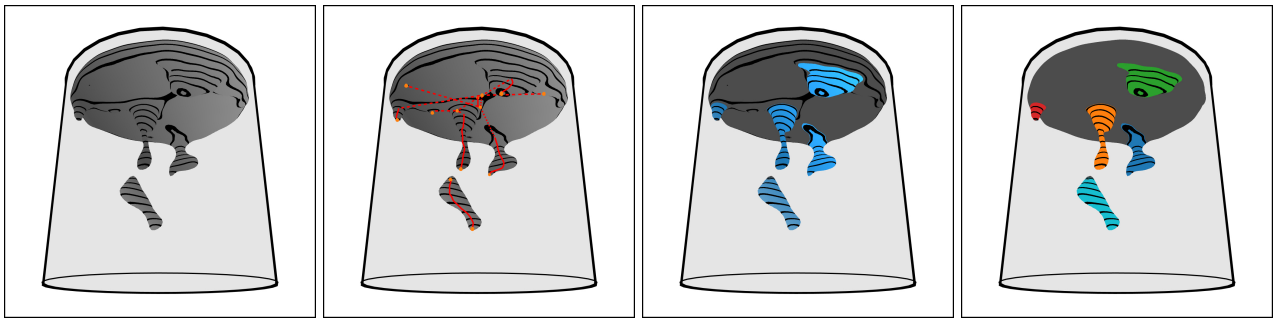
Fig. 2: Particle system with 1,900,000 points rendered with a diverging color map filtered by salt concentration values: all particles (left), particles above concentration 100 (middle), and particles above concentration 200 (right).

Introduction

Viscous fingering is an instability phenomenon that occurs in porous media at the interface between two fluids of distinct viscosity. In particular, it appears when a less viscous fluid is injected within a more viscous one. This process is prominent in many fields of science and engineering [6, 8]; including geology, hydrology, and chromatography. For instance, it plays a key role in the extraction process of oil reservoirs, and in the process of groundwater contamination. The phenomenon is characterized by the formation of a distinctive pattern called viscous fingers (Fig. 1 left). In particular, the geometrical evolution of these patterns provides good indications about the evolution of the penetration of the less viscous fluid. Thus, identifying, tracking, and analyzing the viscous fingers is of first importance for understanding the penetration process. Viscous fingering can be decomposed into three major regimes:

1. Launch: Initially, the interface between the two fluids is approximately planar. Then, the less viscous fluid starts to penetrate the more viscous one when sufficient injection force is applied to it. Viscous fingers of low and uniform amplitude start to appear in this phase.
2. Expansion: Once the reaction is launched, the difference of pressure between the two fluids tends to favor an acceleration of the penetration speed for the areas where the less viscous fluid penetrates the more viscous fluid the most.
3. Termination: Optionally, depending on the characteristics of the media and the fluids, the two fluids can eventually mix together in a termination state, where the finger pattern has completely disappeared, after finger structures merged and dissolved.

We analyze an ensemble of finite pointset method (FPM) simulations that model the viscous fingering process of salt solutions inside water. FPM simulations represent a medium as a numerical point cloud where each point stores properties of the medium such as salt concentration and velocity. The ensemble was provided for the 2016 scientific visualization contest [10] and consists of 50 FPM simulations that differ due to the aleatoric uncertainty of the mixing process. Each member simulates a cylinder that is filled with pure water and contains an infinite salt supply at its top. As soon as the salt mixes with the water, the resulting solutions sink down to the bottom as they have a higher density than the surrounding water. Fig. 2 shows the points of a FPM simulation where each particle has an associated salt concentration value. In course of the simulations the solutions form structures with increased salt concentration value called viscous fingers. However, it is not clear when and where viscous fingers appear and how they evolve. Furthermore, the accuracy of the mean solution rate and properties of the fingers also depend on the point cloud resolution. The ensemble consists of 50 simulations at three resolution levels, i.e., $250k$, $650k$, and $1900k$ particles. Each simulation run provides around 100 timesteps.



a) b) c) d)

Fig. 3: Finger identification pipeline: starting with the voxel volume exceeding the concentration threshold (a), we compute the spatial Reeb graph of the volume (b), then filter the salt supply and noise by using said graph (c), and finally identify connected components (d).

Fig. 3 shows the outline of our approach that is an extension of the methodologies proposed by Aldrich et al. [2] and Favelier et al. [7]. First, we represent the computational domain as a voxel grid and estimate the concentration density distribution for each simulation based on the their particle clouds. Next, we identify viscous fingers as voxels with increased salt concentration density, i.e., superlevel sets in the voxel grid that exceed a certain concentration level (Fig. 3a). To identify individual fingers, noise, and the salt supply, we partition the superlevel sets according to the spatial Reeb graph (Fig. 3c). Each connected component of the partitioned, filtered superlevel set is a single viscous finger (Fig. 3d). To track fingers over time we test their respective volumes for spatial overlaps in time. Finally, we visualize the evolution of the fingers with an interactive tracking graph (Fig. 1 right). Edges of the graph represent the lifespan of individual fingers, and nodes of the graph indicate the time when fingers appear, disappear, merge, and split. The tracking graph is linked to a 3D rendering of the particle data and a direct volume rendered image of the viscous fingers (Fig. 1 left). Users can select fingers in either view to highlight parts of the voxel volume or branches of the tree to interactively explore the evolution of fingers.

We also store for each ensemble member computed metrics—such as size, integral, and bounding box—and direct volume rendered images of the viscous fingers in an image-based ensemble database that was built using the Cinema specification [1]. This database can be searched to compare specific fingers and their evolution. Furthermore, we can use the interactive tracking graph to query the database by selecting important timesteps or fingers in the graph. This enables users to effectively browse, compare, and examine the ensemble in real-time.

Related Work

There exists a huge body of work that demonstrates that tracking graphs are ideally suited to visualize the evolution and relation of components over time [9]. To utilize these methods we first compute a volumetric representation of the point clouds via an adaptation of Shepard's kernel method [13] for three-dimensional domains. Several approaches have been proposed to identify and track components of such time-varying scalar fields. Components can be superlevel sets within the domain that exceed a certain level [12, 15, 5, 14, 11], or subdomains that fulfill geometrical and topological constraints [3, 4]. In our hybrid approach we compute superlevel set components and then filter components based on geometric and topological constraints. Subsequently, the extracted components can be tracked by testing for spatial overlaps in time [12, 5, 11, 15], or applying methods of topological persistence [3, 4, 16]. Specifically, to track the three-dimensional finger structures we extend the tracking algorithm proposed by Lukasczyk et al. [12] for two-dimensional grid cells. We visualize the tracked fingers, their evolution, and the relationship between individual fingers via tracking graphs [3, 5, 12, 16]. Bremer et al. [5] use tracking graphs to visualize burning cells in large-scale combustion simulations where cells

are defined as areas exceeding a fuel consumption rate threshold and are tracked via spatial overlaps. Building on Bremer et al. [5], Widanagamaachchi et al. [16] propose a dynamic version of tracking graphs capable of quick graph layout updates for varying thresholds.

To interactively browse and analyze the ensemble we utilize a Cinema [1] database. In general, such as database consists of pre-rendered images and pre-computed analysis results. The images and analysis results are derived for a range of parameter settings, e.g., camera angle, time, or voxel resolution. Later, a database-viewer can browse through the images and analysis results in real-time as if the results were computed at run time, i.e., emulating interactive data exploration.

Data Preprocessing

The ensemble consists of three sets of simulation runs (one set per resolution). Each run is represented as a time-varying particle system carrying salt concentration where the number, positions, and salt concentration values of particles vary over time. As proposed by Favelier et al. [7], we use an adaptation of Shepard's kernel method [13] to compute for each timestep a volumetric interpolation of the particles onto a 128^3 grid. However, our approach allows the use of other interpolation methods. Next, the regular grid is clipped to the original shape of the domain (a cylinder) to avoid boundary effects of the interpolation. The resulting tetrahedral meshes are then passed as input to our data analysis pipeline. The advantage of this preprocessing is threefold:

1. it brings continuity to the data, which enables the usage of a larger class of analysis algorithms;
2. the input for our analysis pipeline is independent of the particle cloud resolution, and;
3. it allows for interactive renderings of each timestep based on direct volume rendering, isosurfaces, or color mapped clipped views.

Finger Identification

Fig. 3 shows the outline of our detection algorithm. To identify fingers within the voxel data, we start by only considering voxels that exceed a user-specified salt concentration level (Fig. 3a). From our experience, we found that a salt concentration threshold of 10 gave consistent results along timesteps and across simulation runs. We denote with $\mathcal{L}$ the largest connected component of this super-level set. To filter voxels which are considered noise or belong to the salt supply we derive the *spatial Reeb graph* (Fig. 3b) with the method proposed by Lukasczyk et al. [12]. This graph represents the evolution of the fingers in vertical direction (z-direction) for one single timestep, i.e., the graph is a topological skeleton of the fingers and as such captures when fingers form, dissolve, merge, and break into parts while sinking. We use this graph to partition the voxel volume in z-direction and subsequently check for each component of the graph whether its associated voxels have to be filtered out. Specifically, components are identified as part of the salt supply when their xy-bounding boxes are roughly the size of the can and are located at the top of the domain. Valid fingers also have to consists of more than 8 voxels and be at least two grid cells high to filter out small bubbles and bumps, respectively. As shown in Fig. 4, although the *spatial Reeb graph* detects bumps at t_0 and t_2, they only get classified as fingers when they become significantly large enough in timesteps t_1 and t_4, respectively. Those criteria, however, can easily be adjusted according to the interest of the domain scientists. After filtering noise and the salt supply (Fig. 3c), we determine the connected components within the remaining voxels and assign each voxel group a unique, positive integer ID across the entire simulation (Fig. 3d). Voxels not belonging to any group have the voxel group ID -1. Furthermore, for each voxel group (finger) we store its min/max value, bounding box, number of voxels, center of mass, and the salt concentration integral over its volume.

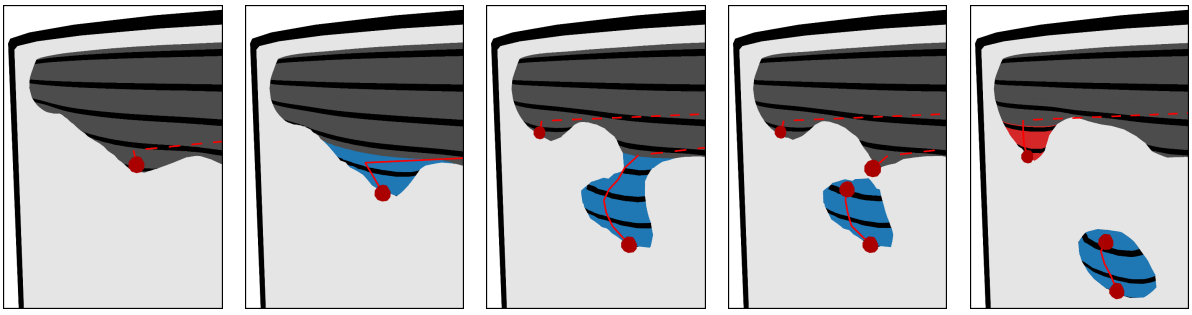


Fig. 4: We use the spatial Reeb graph to filter out noise, the salt supply, and fingers that do not exceed a specified size threshold.

In order to implement this approach efficiently, the concentration values are represented as a 1D floating-point array and the component ID of each voxel is stored in an additional integer array of the same size called the *component matrix (CM)*. Both arrays are passed as *data textures* to a *WebGL shader* that performs direct volume rendering. The shader renders for a given salt concentration the iso-surfaces and colors them according to the *CM* (Fig. 1 left). To enhance spatial perception we also render iso-lines in z-direction, an outline of the computational domain, and use screen-space ambient occlusion. It is also possible to view and superimpose the *spatial Reeb graph* and the underlying particle dataset. The rendering of the particle cloud is interactive, i.e., the analyst can filter particles by salt concentration, adjust the particle sizes, and choose the color map. Moreover, the analyst can examine the mapping between the particles and their associated fingers by color coding them according to the *CM*. By varying the concentration threshold for the finger detection, the analyst is able to examine the impact of said threshold in real time and choose a threshold that seems appropriate to partition the particles into fingers. The computation of the *CM*, the direct volume rendering (512^2 pixels), and the rendering of the particles (512^2 pixels for up to one million particles) is done at interactive framerates even on devices with integrated graphic cards.

Finger Tracking

In order to track fingers over time, we extended the method of Lukasczyk et al. [12] to derive *tracking graphs* for 3D structures. Hence, nodes in the graph represent 3D volumes (the fingers) and nodes of adjacent timesteps are connected by an edge iff their respective 3D volumes overlap. Critical nodes within this graph are nodes with either no previous nodes (birth nodes), no successor nodes (death nodes), multiple predecessor nodes (merge nodes), or multiple successor nodes (split nodes). Hence, a birth node represents the first time a finger is being detected, and a death node marks the last timestep before a finger dissolves. Merge nodes represent the event of multiple fingers joining, while split nodes indicate that fingers break apart.

Fig. 5 shows the *tracking graph* for an entire simulation run where some timesteps are linked to a corresponding DVR image of the fingers. In this illustration of the *tracking graph*, the x-axis represents time, and the y-axis is used to minimize edge crossings to clearly show the evolution of fingers over time by reducing occlusion. Birth and death nodes are shown as discs, while merge and split nodes are shown as diamonds. Each finger is shown in a different color where colors are passed on from the largest predecessor to the largest successor. This makes it easy to follow the history of fingers and links them to the 3D rendering. The thickness of an edge encodes some metric of its corresponding finger, e.g., its average z-position, number of voxels, integral, or length. As demonstrated in Fig. 5, the *tracking graph* effectively illustrates the evolution of the fingers and provides a compact overview of an ensemble member. We also provide an alternative visualization that uses the y-axis of the tracking graph to show a metric of the fingers. For example, in Fig. 6 we use the y-axis to show the average z-position of a finger for the same simulation run as in Fig. 5. This visualization enables users to effectively understand the spatial relationship between fingers. However, this representation is highly occluded, especially when the number of tracked fingers increases. Therefore, it is possible to click on

an edge to highlight all components that are connected to that edge, and fade out the rest. The user can easily switch between both representations and illustrated metrics. By clicking on nodes and edges the user can view information about the associated finger and highlight relevant volumes in the 3D rendering.

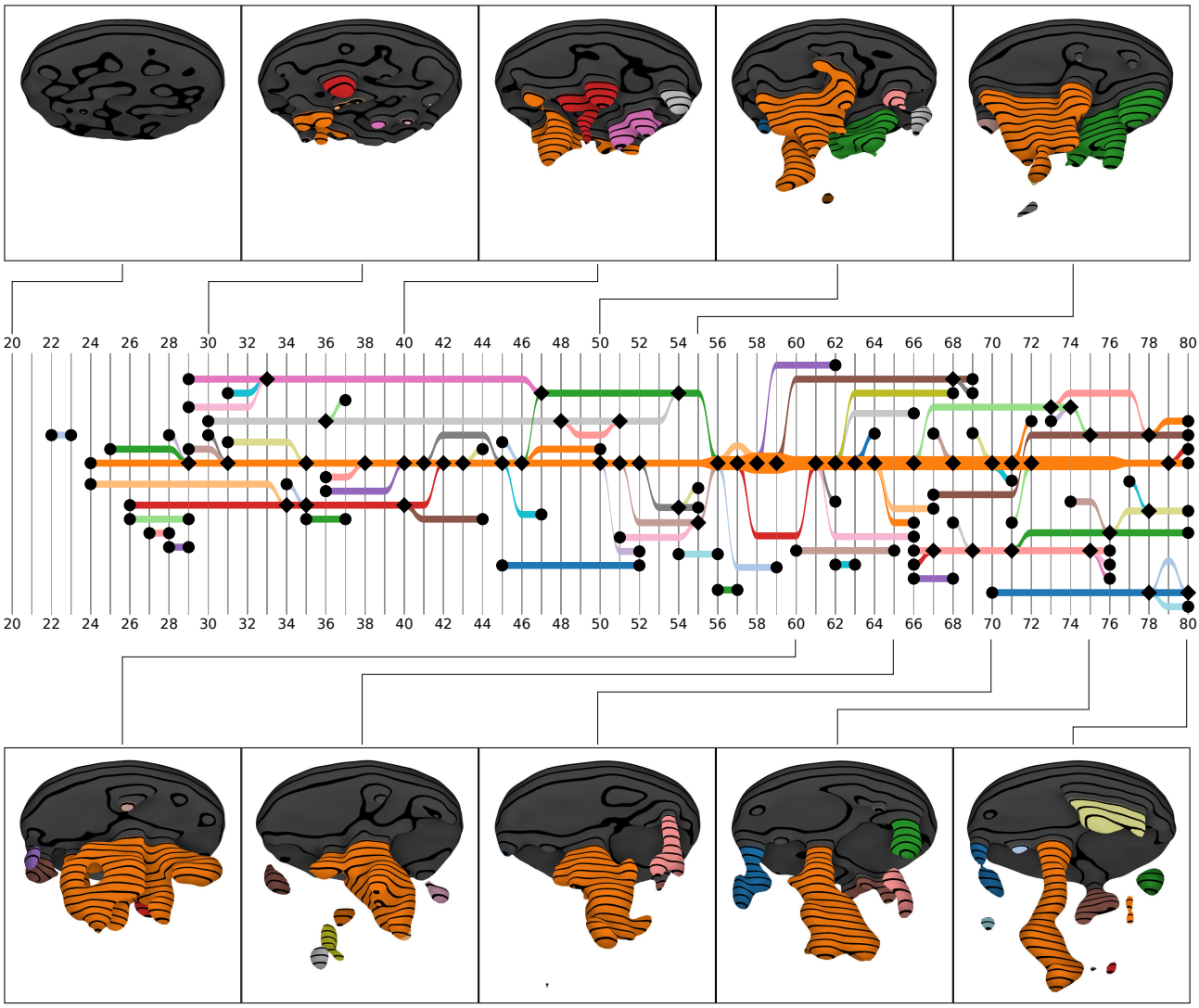


Fig. 5: Overview of one simulation run with $1900k$ particles on a 128^3 grid for salt concentration level 10. As shown by the DVR images (top and bottom), the tracking graph (middle) effectively summarizes the evolution of the viscous fingers (shown in different colors), i.e., the time they appear (discs), disappear (discs), merge (diamonds), and split (diamonds). The x-axis represents time, the y-axis is used to minimize edge crossings, and the line thickness encodes finger volume.

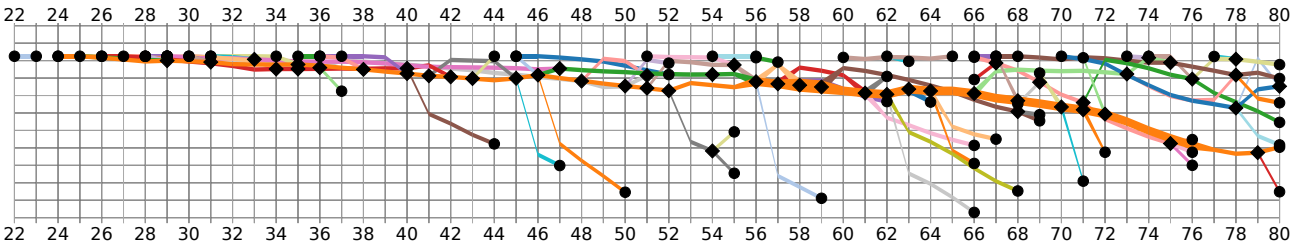


Fig. 6: *Tracking graph* where the x-axis, the y-axis, and the line thickness encode time, average z-position of a finger, and finger volume, respectively.

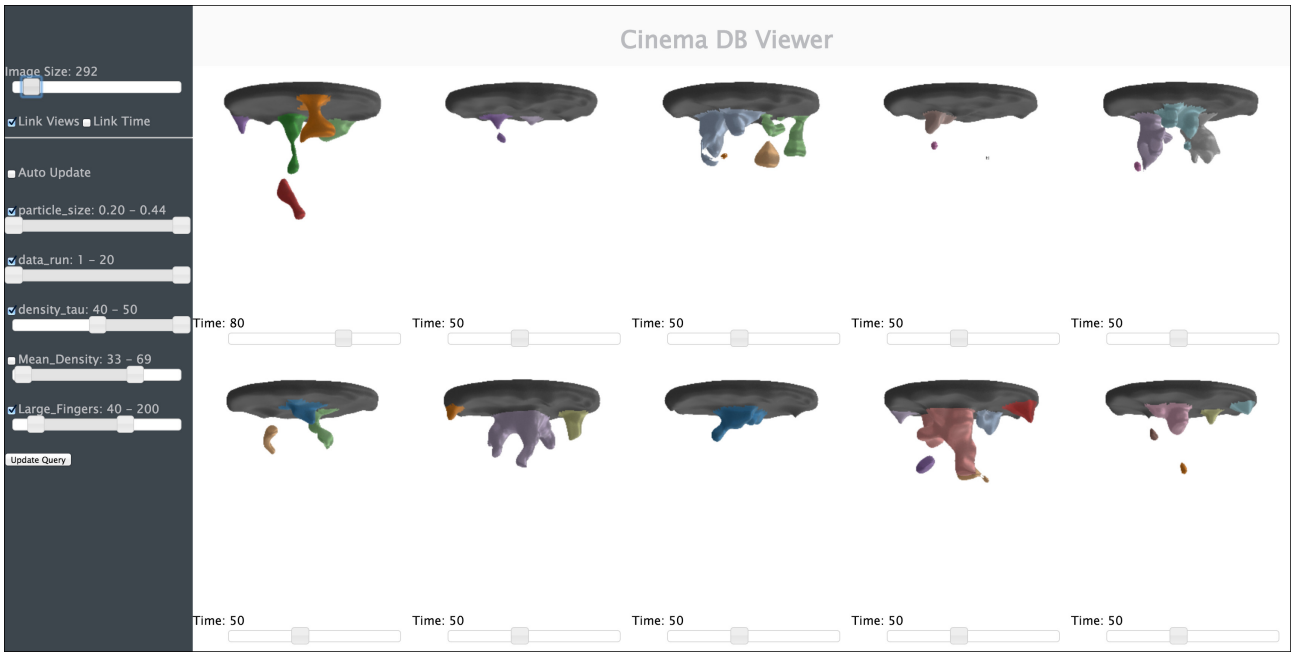


Fig. 7: Cinema view showing a set of images meeting the user's search criteria.

Ensemble Summarization and Comparison

We compare and contrast the ensemble members by leveraging a visualization database framework constructed using the *ParaView Cinema* concept [1]. In this image-based approach, we create a database by analyzing and rendering the dataset with the methods described in the previous sections. Specifically, we store images of the particle dataset and the iso-surfaces for varying camera angles, concentration thresholds, and timesteps. To explore the image database, we implemented a web–based viewer that stores and displays the image database (Fig. 7). Each image in the viewer represents a single simulation; analyzed for a fixed concentration threshold. The camera of an image can be rotated by dragging it with the mouse, emulating real time volumetric interaction. The user can also use a slider to advance the current view over time. For comparison purposes, we also allow the user to sync both the view and the simulation time across all images. The user can browse through the massive image database by using an intuitive query interface. Each image in the database is linked to the parameters used to generate that image and statistical metrics we calculate based on the *tracking graphs* such as the average density, number, lifetime, or length of fingers. Once the user has discovered a dataset of interest, they can explore the simulation in the detail view (Fig. 1).

Using the analysis results stored in the Cinema database we can compute simulation summarizations and comparisons. For instance, to analyze the evolution of the descent of the dissolving salt we can query the minimum z-coordinates of the stored bounding boxes through time (Fig. 8). Three major observations come out of this plot. First, the descent seems linear with time after timestep 10. Second, all runs behave identically in the early timesteps and only little differences can be observed towards the end of the process. The differences that do occur increase linearly over time. Third, the 3 regimes (Launch, Expansion, Termination) are clearly visible: Launch: From timestep 0 to 10, the dissolving salt remains at the top of the domain; Expansion: From timestep 10 to 50-60, the dissolving salt travels down the domain; Termination: All runs hit the bottom of the domain in the same time-frame (timestep 50 to 60), after which a mixing of the two fluids can be observed. To confirm the bounds of the launch regime, we inspect the evolution of the average salt concentration of the dissolving salt $\mathcal{L}$ (Fig. 9). These curves exhibit a clear inflection pattern in the early stages of the simulation, with a clear local maximum at timestep 10 that marks the end of the launch phase.

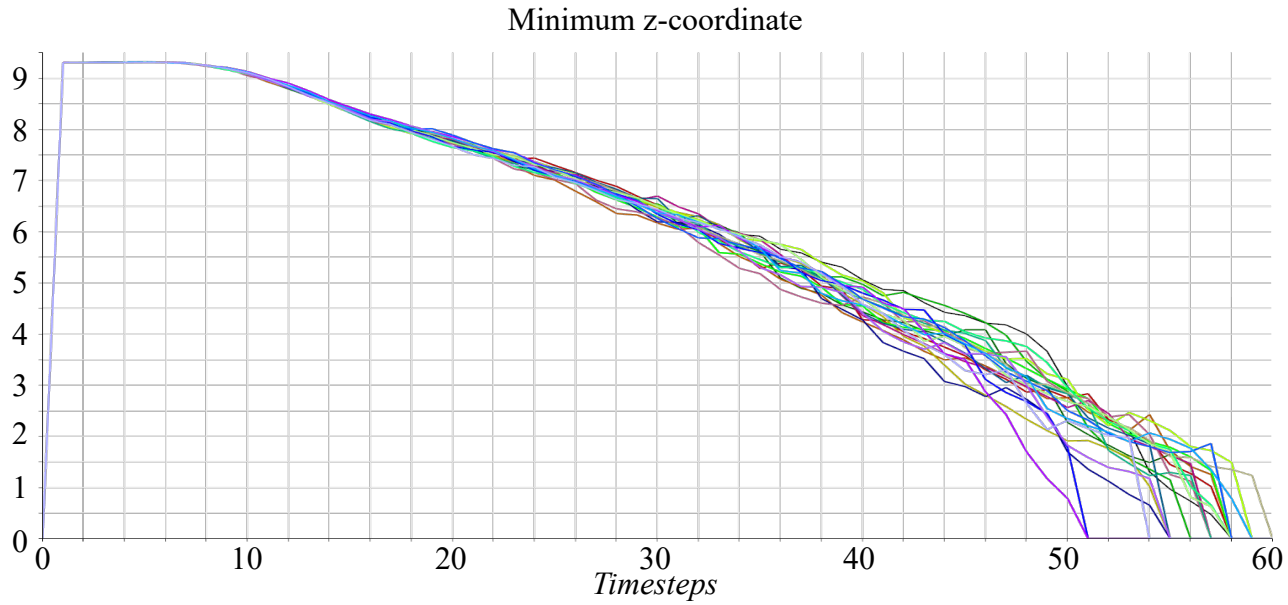


Fig. 8: Evolution of the minimum z-coordinate of the dissolving salt region $\mathcal{L}$ for 22 high resolution simulation runs where each run is shown in a different color. The descent evolution is linear through time (from timestep 10 to timesteps 50-60) and consistent across all runs. For all runs, the dissolving salt is hitting the bottom of the domain in between timesteps 50 and 60.

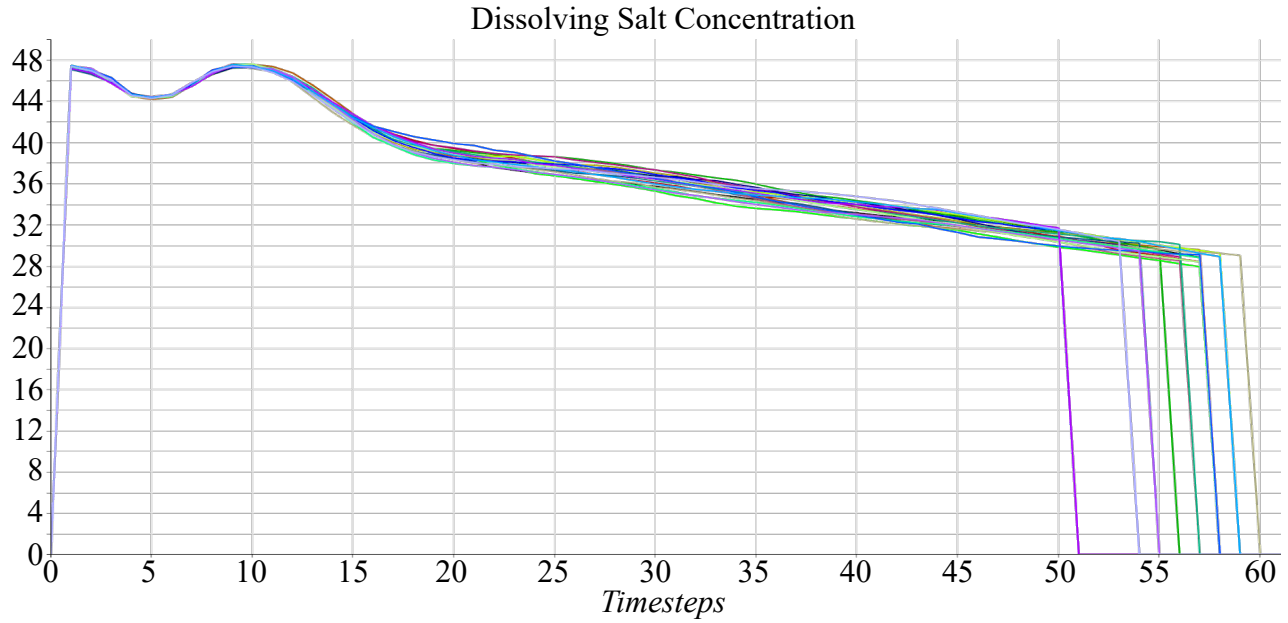
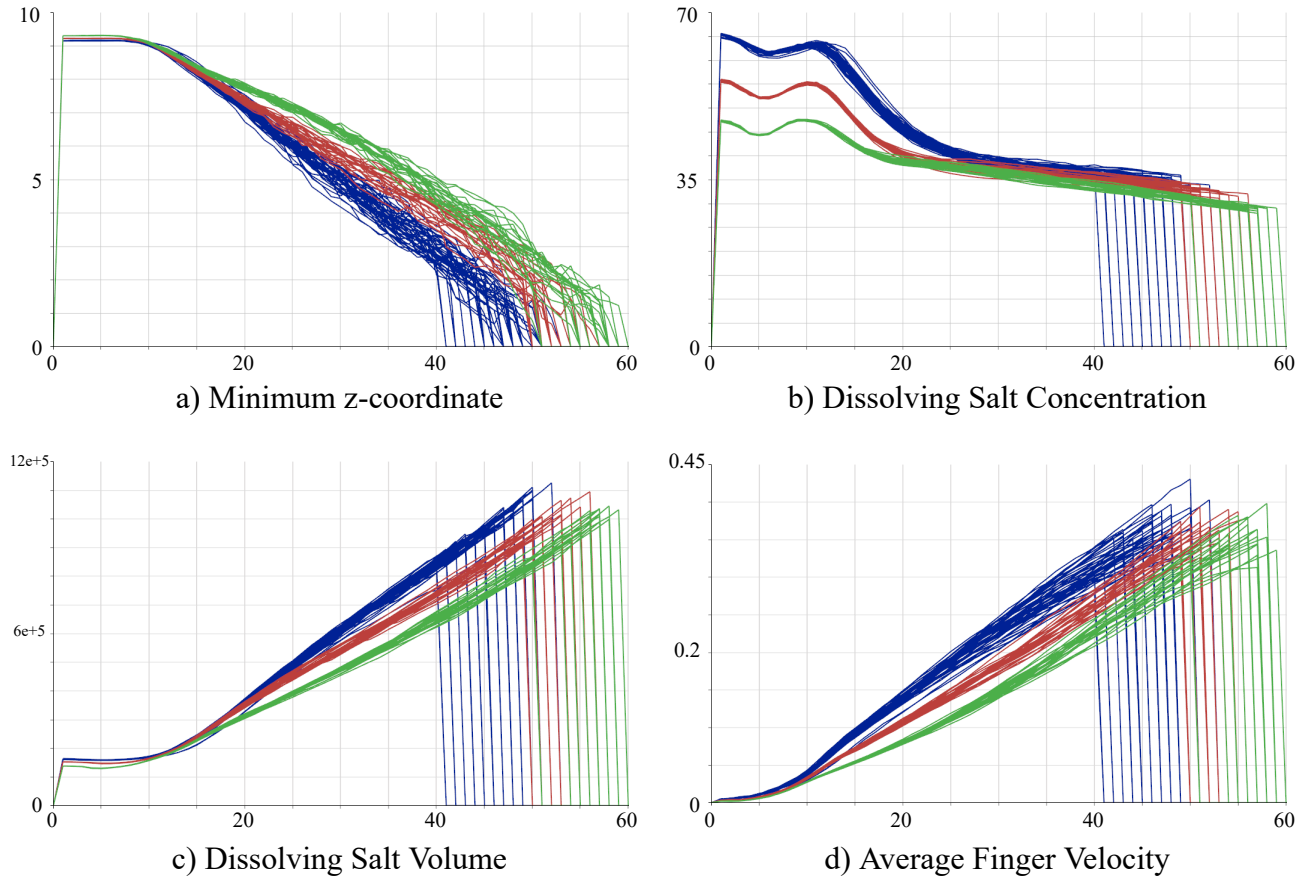


Fig. 9: Average salt concentration evolution of the dissolving salt region $\mathcal{L}$ for 22 high resolution simulation runs where each run is shown in a different color. The curves exhibit a clear inflection pattern in the early stages of the simulation with a clear local maximum of average concentration at timestep 10 that seems to mark the end of the launch phase. The drop of a curve to zero indicates the timestep its corresponding region $\mathcal{L}$ reaches the bottom of the can at which we stop the analysis of the expansion regime.



a) Minimum z-coordinate

b) Dissolving Salt Concentration

c) Dissolving Salt Volume

d) Average Finger Velocity

Fig. 10: Global statistics as a function of time for the 22 high-res (green curves), 23 medium-res (red curves) and 48 low-res (blue curves) runs. These three sets of curves exhibit similar global behaviors: a salt descent according to a linear slope (a), a linear decrease of salt concentration in the expansion regime (b), and a linear increase in both volume (c) and velocity (d). However, it is still possible to distinguish the individual sets as their corresponding curves tend to cluster with only minimal overlap. (a) The minimum z-coordinate of the dissolving salt $\mathcal{L}$ indicates clearly distinct descent rates (different slopes) for each set. (b) The average salt concentration of the dissolving salt $\mathcal{L}$ indicates clearly distinct initial salt concentration levels with a slight delay for the start of the expansion regime (local maximum of concentration). Note that from the middle of the expansion regime onward the decrease rate is similar across resolutions. (c) The volume of the dissolving salt $\mathcal{L}$ also indicates different growth rates (lower resolutions grow faster, as suggested by (a)). (d) The average velocity within the dissolving salt $\mathcal{L}$ also confirms that lower resolution fingers grow faster. The drop of a curve to zero indicates the timestep its corresponding region $\mathcal{L}$ reaches the bottom of the can at which we stop the analysis of the expansion regime. In conclusion, although runs of distinct resolutions exhibit similar global behaviors, each set exhibits distinct initial salt concentrations, delays in the expansion process, and faster penetration rates as the resolution decreases.

To study the impact of the input data particle resolution on the characterization of the viscous fingering process, we visualize global summaries provided by global time-varying statistics for all runs in Fig. 10. This figure plots, for the dissolving salt $\mathcal{L}$, the evolution over time of the minimum z-coordinate, the average salt concentration, the volume, and the average velocity of fingers. In particular, each of the three resolutions is represented with a distinct color. These three sets of curves exhibit similar global behaviors: a salt descent according to a linear slope (Fig. 10a), a linear decrease of salt concentration in the expansion regime (Fig. 10b), and a linear increase in both volume (Fig. 10c) and velocity (Fig. 10d). However, each resolution can be easily distinguished from the others as curves of the same color tend to cluster, with only minimal overlap with the other colors. This indicates clear distinctions between resolutions.

Conclusion and Future Work

We presented a topological data analysis framework adapted to the special case of analyzing finger structures produced across an ensemble of particle based fluid mixing simulations. Our analysis pipeline is based on a volumetric interpolation of the particle data, followed by finger identification and tracking based on geometric and topological constraints. Tracking these finger structures over time by spatial overlaps, we enable the evaluation of relevant time-varying statistics.

We applied our analysis framework on the data-sets provided in the scientific visualization contest 2016 [10]. From this analysis, the viscous fingering process can be characterized into three distinct regimes: launch, expansion, and termination. For these simulations, we were able to bound the occurrences for each of these phenomena. The interactive exploration features of our framework also enabled the quantitative validation of three characterizations within the expansion regime: larger finger structures tend to grow rapidly while smaller structures eventually dissolve. These observations provide insight to the consistent decrease of the numbers of fingers observed as the simulations progress. Additionally, we observed an unexpected behavior, that we named “detachment”, where some fingers suddenly detach from the base of concentrated salt. These detachments typically occurred where the finger structures developed quickly, penetrated deeply, and consisted of a relatively low salt concentration. Our inter-resolution analysis revealed that simulation runs of distinct resolutions had globally similar behavior. However, later expansion starts, faster penetration rates, and fewer viscous fingers were observed as the resolution decreased (fewer particles represented the system as a whole). This comparison shows that the similarity to the highest resolution runs (according to the above criteria) decreases with the resolution, suggesting a convergence of the simulation code for increasing resolutions.

In the future we would like to validate our analysis framework with domain experts. Moreover, we would like to investigate the design of quantitative measures for a deeper evaluation of detachments, as we suspect they may have an impact on the efficiency of the penetration.

References

[1] J. P. 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. *SuperComputing 2014*, pages 424–434, 2014.

[2] G. Aldrich, J. Lukasczyk, M. Steptoe, R. Maciejewski, H. Leitte, and B. Hamann. Viscous fingers: A topological visual analytics approach. *IEEE Scientific Visualization Contest*, 2016.

[3] J. Bennett, V. Krishnamoorthy, S. Liu, R. W. Grout, E. R. Hawkes, J. H. Chen, J. Shepherd, V. Pascucci, and P.-T. Bremer. Feature-based statistical analysis of combustion simulation data. *IEEE Trans. Vis. Comput. Graph.*, 17(12):1822–1831, 2011.

[4] P.-T. Bremer, E. M. Bringa, M. A. Duchaineau, A. G. Gyulassy, D. Laney, A. Mascarenhas, and V. Pascucci. Topological feature extraction and tracking. *Journal of Physics: Conference Series*, 78(1), 2007.

[5] P. T. Bremer, G. Weber, J. Tierny, V. Pascucci, M. Day, and J. Bell. Interactive exploration and analysis of large-scale simulations using topology-based data segmentation. *IEEE Transactions on Visualization and Computer Graphics*, 17(9):1307–1324, 2011.

[6] A. De Wit, Y. Bertho, and M. Martin. Viscous fingering of miscible slices. *Physics of fluids*, 17(5):054114, 2005.

[7] G. Favelier, C. Gueunet, and J. Tierny. Visualizing ensembles of viscous fingers. *IEEE Scientific Visualization Contest*, 2016.

[8] J. Feder. Viscous fingering in porous media. In *Fractals*, pages 41–61. Springer, 1988.

[9] C. Heine, H. Leitte, M. Hlawitschka, F. Iuricich, L. De Floriani, G. Scheuermann, H. Hagen, and C. Garth. A survey of topology-based methods in visualization. *Computer Graphics Forum*, 35(3):643–667, 2016.

[10] IEEEVIS. Scientific visualization contest. http://www.uni-kl.de/sciviscontest/, 2016.

[11] D. Laney, P. T. Bremer, A. Mascarenhas, P. Miller, and V. Pascucci. Understanding the structure of the turbulent mixing layer in hydrodynamic instabilities. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1053–1060, 2006.

[12] J. Lukasczyk, R. Maciejewski, C. Garth, and H. Hagen. Understanding Hotspots: A Topological Visual Analytics Approach. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '15, pages 36:1–36:10. ACM, 2015.

[13] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM National Conference*, ACM '68, pages 517–524, New York, NY, USA, 1968. ACM.

[14] D. Silver and X. Wang. Tracking scalar features in unstructured data sets. In *Visualization '98. Proceedings*, pages 79–86, 1998.

[15] B. S. Sohn and C. Bajaj. Time-varying contour topology. *IEEE Transactions on Visualization and Computer Graphics*, 12(1):14–25, 2006.

[16] W. Widanagamaachchi, C. Christensen, V. Pascucci, and P. T. Bremer. Interactive exploration of large-scale time-varying data using dynamic tracking graphs. In *IEEE Symposium on Large Data Analysis and Visualization (LDAV)*, pages 9–17, 2012.