

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/337671446>

Geometry-Driven Detection, Tracking and Visual Analysis of Viscous and Gravitational Fingers

Preprint · November 2019

CITATIONS
0

READS
65

5 authors, including:



Jiayi Xu
The Ohio State University

10 PUBLICATIONS 70 CITATIONS

[SEE PROFILE](#)



Soumya Dutta
Los Alamos National Laboratory

25 PUBLICATIONS 183 CITATIONS

[SEE PROFILE](#)



Wenbin He
The Ohio State University

10 PUBLICATIONS 32 CITATIONS

[SEE PROFILE](#)



Joachim Moortgat
The Ohio State University

67 PUBLICATIONS 719 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Explainable AI [View project](#)



Urban Data Analysis [View project](#)

Geometry-Driven Detection, Tracking and Visual Analysis of Viscous and Gravitational Fingers

Jiayi Xu, Soumya Dutta, Wenbin He, Joachim Moortgat, and Han-Wei Shen, *Member, IEEE*

Abstract—Viscous and gravitational flow instabilities cause a displacement front to break up into finger-like fluids. The detection and evolutionary analysis of these fingering instabilities are critical in multiple scientific disciplines such as fluid mechanics and hydrogeology. However, previous detection methods of the viscous and gravitational fingers are based on density thresholding, which is sensitive to the user-specified threshold value. Also, the geometric structures of fingers and their evolution are little studied. In this work, we explore the topological and geometric detection and evolution of the fingers in detail to elucidate the dynamics of the instability. We first detect finger cores in three-dimensional (3D) scalar fields to capture the central structures of fingers guided by a ridge voxel detection method; then, we skeletonize finger cores to reveal their overall topological and geometric structures by a Reeb-graph skeletonization method. From the identified finger skeletons, we design a spanning contour tree-based method to capture how fingers branch in 3D space. Then, to display different facets of finger branches, we employ various tree-based visualization methods to plot the finger branches in a plane. Finally, to track the evolution of time-varying fingers and their branches, we nest tree-based geometric glyphs of fingers on a tracking graph. The geometric-glyph augmented tracking graph allows us to study how the fingers grow, merge, and split over time. Feedbacks from earth scientists demonstrate the usefulness of our approach to analyze the evolution of fingers.

Index Terms—Viscous and gravitational fingering, topological and geometric data analysis, ridge voxel detection, tracking graph.

1 INTRODUCTION

IN the context of fluid flow in subsurface porous media (e.g., rock formations), *fingering* refers to flow instabilities when either an invading fluid has a much lower viscosity than the displaced fluid (i.e., viscous fingering), or when a denser fluid flows on top of a lighter fluid (i.e., gravitational fingering). Fingering instabilities lead to a highly non-linear and complex evolution of the displacement front between different fluids [1], [2]. Understanding and predicting flow instabilities is critical in a variety of scientific fields including fluid mechanics [3], [4], computational fluid dynamics (CFD) [5], and hydrogeology [1], [6]. Fingering is generally detrimental when the objective is to displace a viscous fluid (e.g., oil recovery through waterflooding) but can be beneficial when the aim is to mix two fluids, for instance in the sequestration of carbon dioxide (CO_2) [7], [8] in deep water-saturated formations.

In this work, we focus on the latter application, in which gravitational fingering helps to mix dissolved CO_2 throughout the aquifer. This, in turn, helps to guarantee the storage permanence of CO_2 [2], [9]. However, there are chal-

lenges for detection and evolutionary analysis of viscous and gravitational fingers. We illustrate the challenges and high-level motivations of this work from two perspectives: (1) domain-specific requirements obtained from an expert, and (2) limitations of previous viscous and gravitational fingering analysis techniques. Below we expand on each of these.

We involved an expert in the field of Earth Sciences to help us comprehend the challenges and domain-specific requirements in the detection and visualization of viscous and gravitational fingers. Uniquely identifying fingers is challenging because fingers are unstable structures in the fluids, and result from complex fluid interactions. However, flow instabilities, whether in the subsurface or space, have recently been found to obey specific universal scaling laws (e.g., [2], [9]). Such scaling laws can be used to estimate the severity and evolution of flow instabilities for different sets of conditions, without having to redo high-resolution, and thus computationally expensive, simulations. To reveal those scaling laws, the analysis of pattern formations is critical, i.e., the onset, growth, merging, and splitting of fingers and the connectivity between fingers in space and time. The earth scientist pointed out that tracking and quantifying these features in a 3D volume, to the best of his knowledge, is not possible with any standard visualization tools.

Extracting fingers is challenging, and different techniques have been proposed [5], [6], [10], [11], [12], [13]. However, the existing detection techniques use a specific density or concentration threshold to filter the data first, which may lead to the loss of low-density branches, and also regard each finger as a single entity without further analyzing its internal geometric structures (e.g., branches), which is essential to the scientists in understanding flow instabilities [14], [15], [16], [17].

- J. Xu is with the Department of Computer Science and Engineering, The Ohio State University, Columbus, OH, 43210.
E-mail: xu.2205@osu.edu
- S. Dutta is with the Science at Scale Team, Los Alamos National Laboratory, Los Alamos, NM 87545.
E-mail: sdutta@lanl.gov
- W. He is with the Department of Computer Science and Engineering, The Ohio State University, Columbus, OH, 43210.
E-mail: he.495@osu.edu
- J. Moortgat is with the School of Earth Sciences, The Ohio State University, Columbus, OH, 43210.
E-mail: moortgat.1@osu.edu
- H.-W. Shen is with the Department of Computer Science and Engineering, The Ohio State University, Columbus, OH, 43210.
E-mail: shen.94@osu.edu

Manuscript received February 26, 2019; revised September 27, 2019.

To satisfy the requirements of scientists and incorporate geometric structures of viscous and gravitational fingers during the visual analysis, in this study, we propose a geometry-driven solution for finger extraction and visualization. We first model central regions of fingers as ridges of 3D space and identify finger cores by applying a ridge voxel detection method. From finger cores, we obtain the complete volume of fingers in the 3D scalar fields, and produce finger skeletons by using a Reeb graph skeletonization to acquire overall geometric features of branching fingers in 3D space. To construct finger branches, we propose a technique to generate spanning contour trees [18] from finger skeletons. Next, we visualize fingers and their branches for all timesteps by a geometric-glyph augmented tracking graph so that we can track and compare them efficiently. Given the resulting branches of each finger, various tree-based visualization glyphs are nested into the tracking graph to display the geometric structures of fingers and branches. To reveal connections between finger branches with minimal crossings, we employ a tree-drawing method [19] to plot the side view of fingers. To contrast the complexity of finger branches and preserve their relative positions, we use a spatially ordered treemap technique [20] to produce rectangular glyphs to draw the top view of fingers. Hence, our contributions are twofold:

- 1) For extraction of finger cores from three-dimensional scalar fields, we propose a ridge voxel detection technique based on Taylor's theorem, which is inspired by the detection of pixels with ridge points in [21]. For detection of finger skeletons, we use a Reeb graph skeletonization [22]. Furthermore, to construct branches of each finger, we provide a technique to produce spanning contour trees [18] from finger skeletons.
- 2) We provide an interactive visual-analytics system which allows efficient exploration of fingers over space and time with minimized occlusion. Our system offers a new geometric-glyph augmented tracking graph design which reveals the temporal evolution of the fingers and their branches effectively.

2 APPLICATION BACKGROUND

In this section, we elaborate on the concept of fingers and how they form, discuss the domain-specific requirements, and summarize the limitations of previous analyses of viscous and gravitational fingers.

2.1 Viscous and Gravitational Fingers

Viscous and gravitational flow instabilities in porous media result in finger-like features (e.g., Fig. 9d) visually, and the fingering phenomenon refers to the formation and evolution of such fingers. The instabilities are triggered by adverse mobility or density ratios between displacing and displaced fluids [1]. *Viscous fingering* is caused by viscosity contrasts between fluids: when a less viscous fluid is injected into a more viscous medium, the less viscous fluid tends to penetrate through the more viscous fluid to form elongated finger-like structures [3]. *Gravitational fingering* refers to instability caused by contrasts in density between fluids: when a denser fluid resides (e.g., is injected) on top of a less

dense fluid, the interface (or rather, a diffusively growing boundary layer) can become unstable; also, the denser fluid vertically penetrates the lighter fluid, while the lighter fluid rises buoyantly.

Data description. The fingering dataset that we used in this work was generated from simulations of injecting carbon dioxide (CO_2) from the top of a water-saturated reservoir. In our application, the gravitational fingering helps to mix injected CO_2 throughout the aquifer. When CO_2 dissolves in water, it locally increases the water density in the top, which is prone to gravitational instabilities. The CO_2 -water front can become unstable and lead to fingering of CO_2 -enriched, denser, water downwards with upwelling of fresh lighter water. The time-varying datasets have more than one hundred timesteps for each simulation. The spatial data of each timestep is defined within a cuboid reservoir, which is 40 meters in height and has a 30 meters \times 30 meters base. The reservoir is represented by a 3D rectilinear grid (90 \times 90 \times 100) with uneven spacing, where each grid cell is corresponding to a CO_2 density value. Fingers in our data can be identified as regions with local high-density of CO_2 .

2.2 Requirements of the Domain Expert

First, we discuss the application-specific requirements that were identified by our Earth Sciences domain expert, who has ten years of experience in researching fluid injection processes and associated viscous and gravitational fingering instabilities. In his day to day studies, the expert often visually analyzes the fingers using visualization tools such as ParaView [23], VisIt [24], and Tecplot [25]. During our interactions, the expert mentioned that he was interested in how to visualize the 4D (3D in space plus time) simulations of fluid injection. He stated that, without the ability to look at the data in many different ways efficiently, it was often hard to even know what questions to ask or answer. The expert informed us that the current visualization techniques that he used were not ideal. For example, some domain tools visualize fingers as hollow sheets rather than dense columns; also, the three-dimensional fingers which were visualized by standard visualization methods such as volume rendering and isosurfaces suffered from the occlusion problem. Thus, the expert usually had to cut multiple cross-sections of those fingers to analyze the formation and internal structures of fingers rigorously. However, tracking and quantifying the 3D geometry of these ramified structures in both space and time is virtually impossible with those methods, i.e., how hundreds of distributed fingers grow vertically, spread horizontally, shield and merge at certain timesteps, and then predominantly split into new smaller fingers at other timesteps. Visualizing and quantifying these geometric features provide insights into the underlying physics such as the analysis of bimolecular reaction [26] and physical flow regimes of, e.g., enhanced or suppressed mixing rates [4], [8]. Also, the geometrical features including widths [7] and tip locations [7] of fingers have been used to analyze scaling behaviors. However, the vast majority of studies [7], [26] on earth sciences for geometric analysis of fingers are in 2D and often that has been done essentially by hand and visual inspection. He thought that interactive space-time diagrams of the fingers, which can effectively

present the finger-specific events such as merging, splitting, and branching of fingers, would be extremely valuable.

2.3 Limitations of Existing Works for Fingers

In this section, we review the methods used previously to detect and visualize viscous and gravitational fingers, and highlight the limitations of the existing methods.

2.3.1 Limitations of Current Detection Methods of Fingers

To detect fingers and their branches, domain scientists require a robust detection method. Since the formation of fingers is extremely non-linear, accurate detection of the fingers is a non-trivial task. To extract fingers from the density (or concentration) scalar fields, previous researchers typically used thresholding on the density value to extract regions with high density and interpreted the connected (or, close) regions as fingers of interest. In the previous study, Skauge et al. [6] and Fu et al. [10] modeled fingers as high-density vertical lines, which were detected by a peak detection method. However, fingers are complex structures with branches rather than simple vertical lines. Aldrich et al. [11] and Lukasczyk et al. [13] considered each finger to be a connected component of high-density 3D regions and detected fingers through connected components. In an alternative approach by Favelier et al. [12], instead of detecting the fingers directly, they detected the fingertips first and segmented the volume by a watershed traversal method to obtain the full fingers from the fingertips. Luciani et al. [5] studied particle data and grouped particles with close locations and concentrations to be fingers. However, these methods [5], [11], [12], [13] usually need a specific density or concentration threshold to filter the data first. Even though they obtained good results, the use of such hard thresholding can make the finger extraction sensitive to the threshold value used. For example, some fingertips (e.g., a bottom part in Fig. 5a) have much lower densities than the finger roots (i.e., top parts of the fingers). If the threshold is too large, these fingertips that do not satisfy such threshold may be missed. If the threshold is too small, different fingers cannot be segmented because regions between fingers also have densities such as Fig. 6b. Lukasczyk et al. [27] also reported that it is not sufficient to partition fingers based on a certain density threshold. Furthermore, these methods typically concentrate on detecting each finger as a single entity, but our application study needs more detailed branching information of fingers, which the majority of the above works have not explored.

2.3.2 Limitations of Current Tracking Graphs of Fingers

When it comes to the visualization of time-varying fingers, none of the existing tracking visualizations focus on the geometric evolution of the fingers. The tracking graphs provided by Aldrich et al. [11] display fingers at each timestep as points in a column. Aldrich et al. used hues to encode different fingers and linked related fingers of adjacent timesteps by curves. In a recent work, Lukasczyk et al. proposed nested tracking graphs [27], which can depict the evolution of level-sets of density fields and visualize the evolution of fingers across multiple specified density thresholds. The lack of tracking graphs for the geometric evolution

of fingers which encodes the branching information and is not sensitive to the specified density thresholds further motivates us to develop a new geometry-driven tracking graph for the analysis of time-varying fingers.

3 RELATED WORK

We discuss related techniques to our geometry-driven approach.

Ridge detection methods. Ridges originally are structures of surface topography whose mathematical properties were studied by de Saint-Venant [28]. In mathematics, the ridges of a smooth function are a set of curves or surfaces whose points are local maxima of the function in a certain number of dimensions; specifically, ridge lines are curves whose points are local maxima in $n - 1$ dimensions if the dimensionality of the whole space is n . The concept of ridges has been used to detect and extract curvilinear structures in computer vision [21], [29] and scientific visualization [30], [31]. In computer vision, we can regard the grey-level image as a 2D scalar field and extract curvilinear structures such as roads [21] and human fingers [29] by applying the ridge line detection method. In scientific visualization, definitions of ridges are extended to volumetric data making ridges a valuable tool for scientific visualization [30], [31]. However, conventional ridge line detection methods [21], [30], [31] may not be stable for real datasets. Kern et al. [32] reported that, in their 3D jet stream application, eigenvectors computed by the classic ridge extraction techniques are unstable, leading to detected ridge lines that are broken and cluttered.

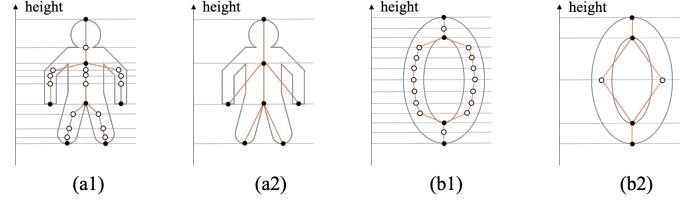


Fig. 1: This figure shows how to extract the skeleton of a 2D human-like object and an O-shaped object by the Reeb-graph-based method intuitively. We first shrink the level-sets of the height function to get intermediate results for the two objects in (a1) and (b1). Solid circles represent critical points of the height function; hollow circles represent passing points. Next, we remove the intermediate passing points and connect the critical points with straight lines to simplify the intermediate skeletons to obtain the Reeb graph embeddings in (a2) and (b2). Note that the O-shaped object contains a loop, so we preserve two passing points for the loop structure in (b2).

Reeb graph skeletonization. The Reeb graph method is one of the standard skeleton-extraction approaches to approximate skeletons of objects [33]. Fig. 1 is an example to derive skeletons of 2D objects based on Reeb graphs. By following the evolution of the level-sets of a height function defined on a compact manifold, the Reeb graph obtains the topology of this manifold; nodes in the Reeb graph represent critical points of the height function, and edges correspond to connections between critical points [33], [34]. The resulting Reeb graph is not a skeleton; however, Lazarus and Verroust [22] embedded the resulting Reeb graph into

the 3D space to get the skeleton of the original object, which can extract the skeleton of an object accurately. If a 3D object has no holes inside, the contour tree [35], as a special instance of the Reeb graph, is sufficient for the skeletonization of this object such as the example provided in the Topology Toolkit (TTK) [36].

4 APPROACH OVERVIEW

A brief overview of our solution is presented in Fig. 2 through a schematic diagram. In this work, we focus on the study of geometric structures of fingers and propose several techniques to visualize the temporal evolution of fingers and branches with minimized occlusion. From the raw density scalar fields, we apply a ridge voxel detection method to extract central structures of fingers as finger cores and skeletonize finger cores into finger skeletons by a Reeb graph skeletonization. Finger skeletons provide the overall geometric structures of fingers. The volume of each finger is extracted from scalar fields based on the finger cores; then, we offer volume visualization techniques to show the finger in the spatial domain. Also, we project finger skeletons onto a plane to observe spatially relative positions between fingers. We also construct finger branches from the finger skeletons using a variant of spanning contour tree method. Given the branches of fingers, we adopt various tree-based visualization methods to represent different aspects of fingers' branches. Finally, to track the geometric evolution of time-varying fingers, we augment a tracking graph with the geometric representations of fingers which allows us to study how the fingers change, merge, and split over time. On the geometric-glyph augmented tracking graph, fingers between consecutive timesteps with volume overlapping are connected by links for temporal tracking; branches with volume overlapping can be identified by interactions.

5 GEOMETRY-DRIVEN DETECTION OF VISCOUS AND GRAVITATIONAL FINGERING

In this section, we provide details of the proposed geometry-driven techniques used for the detection and tracking of viscous and gravitational fingers.

5.1 Extraction of Fingers

We propose a finger extraction method for density fields. Since fingering is highly non-linear and the shapes of resulting fingers are varying, it is hard to define fingers by mathematical expressions directly. Instead, we model finger cores, the central regions of fingers, as ridges. Then, the complete fingers are defined by the finger cores plus the connected lower-density regions covering the finger cores. In contrast to previous methods that extract fingers based on hard thresholding of the density, our method first extracts the core structure of fingers guided by a ridge voxel detection method, then, skeletonize finger cores into finger skeletons which are linear structures with geometric information. Finally, we expand the finger cores to complete volume of fingers. As a result, our approach is not sensitive to a density threshold value, and low-density finger branches can be preserved.

5.1.1 Ridge Voxel Detection Guided Extraction of Finger Cores

We extract central structures of fingers to be finger cores by detecting voxels that contain ridges to capture regions with locally maximal densities. In this work, we model finger cores as structures that are at the center of finger branches and have higher densities locally. This modeling is based on the following reason. According to the earth scientist, the fingering is a complex process. One of the causes of fingering is the diffusive process. The diffusion of fingers is driven by compositional gradients (i.e., the difference of CO₂/water composition). Through the diffusive process, CO₂ volume tends to spread out from high-density regions to low-density regions; the regions with higher densities usually become the center of finger structures. For example, when fingers grow faster than CO₂ is supplied from the top, the density contrast decreases and fingers may stagnate; once they 'fill up' with fresh CO₂ from the top, the density contrast increases and the fingers can grow again. Also, when CO₂ accumulates in a finger tip, this can lead to so-called tip-splitting, where a finger splits/branches into multiple fingers.

To extract finger cores, conventional ridge line detection methods [30], [31] may not be appropriate for two reasons.

- 1) Traditional ridge line extraction methods detect precise paths for ridge lines across voxels. In our application, exact paths of ridge lines may not be proper because, according to the study of Damon [37], ridge lines cannot express branching structures well, which may lead to disconnected branching structures of fingers. Our proposed method detects ridge voxels instead of precise paths of ridge lines, which preserves the necessary branching structures of fingers.
- 2) If multiple ridge lines pass through same voxels, traditional methods separate the distinct ridge line pieces in the shared voxels, which is not appropriate in our application. Because the CO₂ volumes containing these ridge line pieces overlap at the shared voxels, these CO₂ volumes interact with each other at the shared voxels. The interactions between CO₂ volumes usually are corresponding to the branching of finger structures. Hence, unlike the traditional methods where ridge lines with shared voxels are considered separate, we group such ridge lines into the same finger structure.

To obtain regions with higher densities locally, we filter the 3D domain to identify voxels (i.e., grid cells) containing ridges, which are named as *ridge voxels* in the paper. When extracting 2D curvilinear structures, Steger [21] used Taylor's theorem to locally approximate the 2D image function centered at each pixel by Taylor polynomials to detect pixels having ridge points. Inspired by Steger's 2D detection method [21], in the 3D space, we determine whether a given voxel contains ridge points based on Taylor's theorem. The produced results are illustrated in Fig. 3 and Fig. 4.

We discuss details of our ridge voxels extraction method, which determines whether a given voxel of a 3D density field contains points on ridge lines or not based on Taylor's theorem. Let center position of the given voxel be $p_0 = (x_0, y_0, z_0)^T$. Let r_x, r_y , and r_z be the locally estimated first-

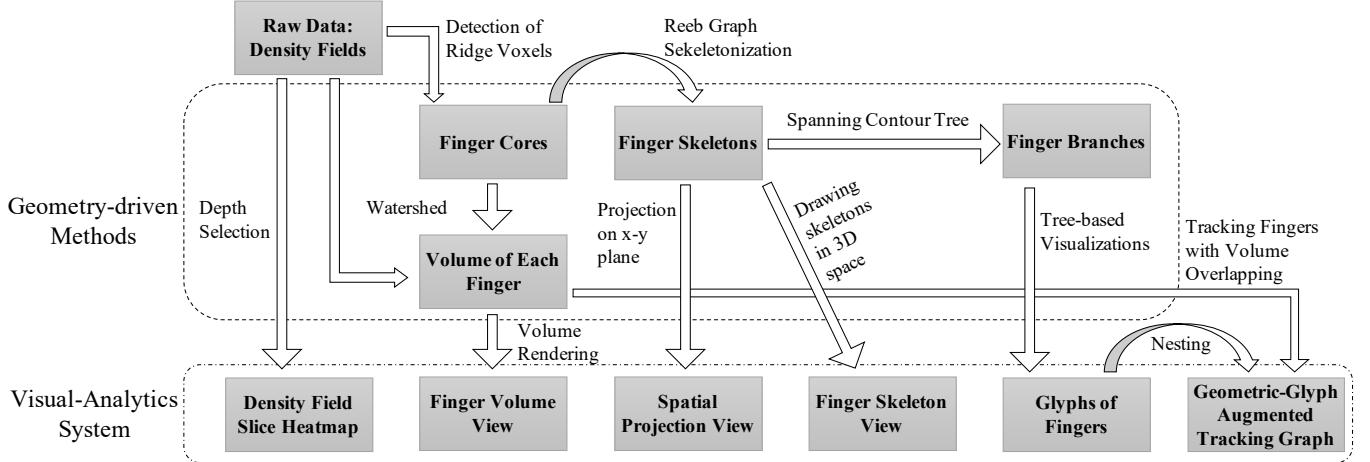


Fig. 2: Schematic diagram of our application for viscous and gravitational fingering.

order derivatives at the center location. Then, the Jacobian matrix at the center position is given by

$$J(\mathbf{p}_0) = J(x_0, y_0, z_0) = (r_x, r_y, r_z)^T. \quad (1)$$

Hessian matrix at the center position is given by

$$H(\mathbf{p}_0) = H(x_0, y_0, z_0) = \begin{bmatrix} r_{xx} & r_{xy} & r_{xz} \\ r_{xy} & r_{yy} & r_{yz} \\ r_{xz} & r_{yz} & r_{zz} \end{bmatrix} \quad (2)$$

We use the second-order multivariate Taylor polynomial to approximate the density function locally. Let $f(\mathbf{p})$ be the local approximation for the density function of any point $\mathbf{p} = (x, y, z)^T$ within the voxel, and let $\Delta\mathbf{p} = \mathbf{p} - \mathbf{p}_0$. The $f(\mathbf{p})$ is expressed by:

$$f(\mathbf{p}) = f(\mathbf{p}_0) + J(\mathbf{p}_0)^T \Delta\mathbf{p} + \frac{1}{2} \Delta\mathbf{p}^T H(\mathbf{p}_0) \cdot \Delta\mathbf{p}. \quad (3)$$

The first-order derivatives of $f(x, y, z)$ are included in the Jacobian matrix, which is

$$J(\mathbf{p}) = (f_x(\mathbf{p}), f_y(\mathbf{p}), f_z(\mathbf{p}))^T = J(\mathbf{p}_0) + H(\mathbf{p}_0) \cdot \Delta\mathbf{p} \quad (4)$$

The Hessian matrix of the second-order Taylor polynomial is the same everywhere $H(\mathbf{p}) = H(\mathbf{p}_0)$.

To estimate the ridge points within the voxel, we refer to the criterion of ridge points. For any point $\mathbf{p} = (x, y, z)^T$ within the voxel, among all eigenvectors of this point, let two eigenvectors, whose eigenvalues are top two maximal absolute values of this point, be $\vec{v}_1 = (v_{1x}, v_{1y}, v_{1z})^T$ and $\vec{v}_2 = (v_{2x}, v_{2y}, v_{2z})^T$. The first-order directional derivatives along the two eigenvectors for the point \mathbf{p} are

$$d_i(\mathbf{p}) = \vec{v}_i^T J(\mathbf{p}), i = 1, 2. \quad (5)$$

Based on Lindeberg's criterion [29], this point is a ridge point if and only if the two first-order directional derivatives are zero and the second-order directional derivatives along the two eigenvectors are smaller than zero. Note that, for 2D space, to detect the ridge point within a pixel conveniently, in addition to the basic ridge criterion [29], [38], Steger's method [21] has one additional assumption: the ridge point should lie on the line which passes the center of a pixel and is at the direction of the eigenvector with the maximal absolute eigenvalue. However, this assumption brings more restriction to the detection of ridge voxel in the 3D space

and may lead to missing ridge voxels. Hence, different from Steger's method [21], our approach uses the basic ridge criterion [29], [38] solely to derive inequalities to determine whether a voxel contains ridge points or not.

Next, we examine whether there exists a ridge point $\mathbf{p}_1 = (x_1, y_1, z_1)^T$ within the given voxel. We obtain the two first-order directional derivatives at the ridge point \mathbf{p}_1 by $d_1(\mathbf{p}_1)$ and $d_2(\mathbf{p}_1)$ and let the two derivatives be zero. Note that we have two equations (i.e., $d_1(\mathbf{p}_1) = 0$ and $d_2(\mathbf{p}_1) = 0$) and three unknowns (i.e., x_1 , y_1 , and z_1) hence we can get expressions of two of the unknowns by polynomials of the rest unknown. Without loss of generality, we represent x_1 and y_1 by polynomials of z_1 . Next, we constrain that the ridge point $\mathbf{p}_1(x_1, y_1, z_1)$ is within or on boundary of the voxel to obtain inequities of x_1 , y_1 , and z_1 . We substitute x_1 and y_1 by the polynomials of z_1 to get the inequities only associated with z_1 . Based on the inequities of z_1 , we can determine whether z_1 has the feasible value such that the ridge point \mathbf{p}_1 exists within or on the boundary of the voxel. Finally, we get second-order directional derivatives at \mathbf{p}_1 along \vec{v}_1 and \vec{v}_2 based on the Hessian matrix $H(\mathbf{p}_1)$ to ensure that the second-order directional derivatives are less than zero for this voxel to be declared a ridge voxel.

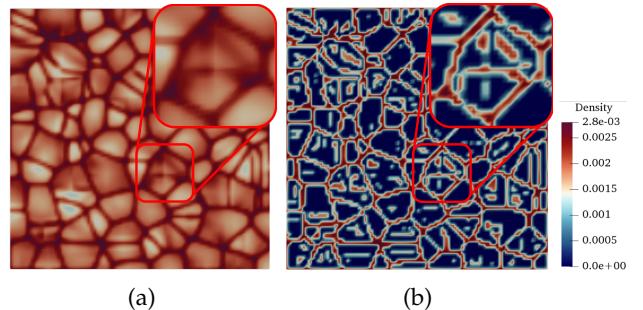


Fig. 3: We show the top view of (a) the raw 3D space and (b) the extracted ridge voxels. As the same places are highlighted in (a) and (b) by red, our ridge-guided method can detect high-density hexagonal cells in the top grid cells and high-density curvilinear structures inside the hexagonal cells.

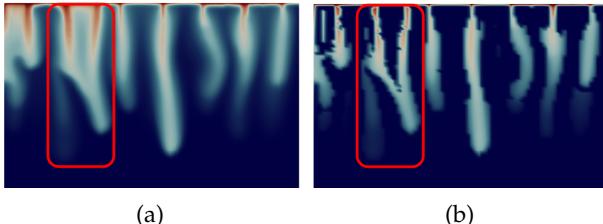


Fig. 4: We show the side view of (a) the raw 3D domain and (b) the extracted ridge voxels (i.e., finger cores). As the same finger is highlighted in (a) and (b) by red, our ridge-guided method captures the geometric structure of a finger with fusion and splitting of branches.

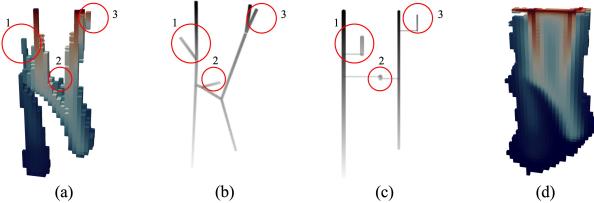


Fig. 5: For illustration, we display a simple example of an extracted finger by four images: (a) the finger core (i.e., the extracted ridge voxels), (b) the finger skeleton, (c) the linear glyph of the finger, and (d) the complete finger volume (i.e., all the voxels of the finger). Three small branches are circled by red and labeled with numbers for readers to identify the correspondence conveniently.

5.1.2 Reeb Graph Based Skeletonization for Finger Skeletons

We skeletonize the extracted ridge voxels to obtain finger skeletons by Reeb graph skeletonization method [13], [22], [33], [34]. In computational geometry, Reeb graphs are well-known for their ability to show the skeletons of a 3D object [33] and preserve branches and heights of a 3D object accurately [34]. Another approach of shape skeletonization is based on the contour tree, which is used in TTK [36]. However, in our application, voxels that are not ridges are removed, which may leave holes in the volume of fingers (e.g., Fig. 8a); the volume of fingers may have torus-like structures. Hence, the contour-tree-based method is not appropriate for our application. Fig. 1 displays an example where skeletons are extracted from a 2D human-like object and an O-shaped object by the Reeb graph method. The intermediate skeletons (e.g., (a1) and (b1)) contain all the geometric information of the object; the final results (e.g., (a2) and (b2)) usually only contains critical points of the height function and edges that connect the points, which provides simplicity and clear representations to visualize the necessary branching and height information. If one requires to preserve more geometric information such as the curvature and length of branches, the intermediate skeletons (e.g., (a1) and (b1)) can be directly used.

We generate the Reeb graphs from the detected ridge voxels of the fingers. We sweep ridge voxels of fingers vertically and shrink the connected components to points to form its Reeb graph. The resulting Reeb graphs are then embedded in the 3D spatial domain to obtain skeletons of the fingers by preserving its branching and height information. The resulting skeleton of a simple finger is shown in Fig. 5b. A complex example is shown in Fig. 7 and Fig. 8d and simplified in (e). Compared with finger volume,

finger skeletons make the interpretation of the branching information intuitive.

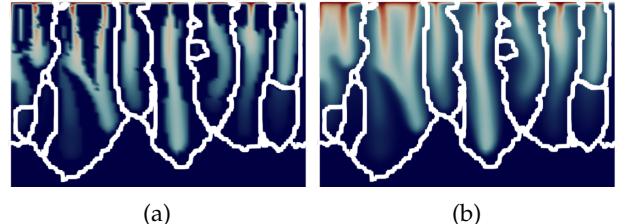


Fig. 6: We show the segmentation boundaries by white lines on (a) the extracted finger cores and (b) the raw 3D domain.

5.1.3 Segmentation of Finger Volume in Spatial Domain

To recover each of the complete finger structures in the 3D spatial domain from the detected ridge voxels, we segment the density field into connected subfields. In computer vision, the watershed technique is well-known for segmenting images. In a previous study, Favelier et al. [12] segmented the volume of a scalar field by the watershed traversal method [39] to capture fingers given fingertips. Following a similar strategy, we use a flood-filling watershed algorithm to extract the fingers in the 3D domain from the detected ridge voxels; intuitively, a given unclassified voxel will be classified into a finger containing the nearest ridge voxel to the unclassified voxel. In the following, we list steps to perform the segmentation.

Step 1 We group ridge voxels to be connected components.

Each connected component of ridge voxels forms a single finger, and each finger is assigned a unique index. Note that, almost all fingers are connected through the high-density hexagonal cells in the top layer (i.e., the diffusive boundary layer contained in the top grid cells), for example, Fig. 3. Hence, to separate fingers, as suggested by the previous researchers [5], [11], [13], we ignore the ridge voxels of the top layer during this grouping step.

Step 2 We label ridge voxels of each finger by the finger's index and insert these ridge voxels into a queue.

Step 3 We get a labeled voxel from the queue and assign the finger index of the labeled voxel to its unlabeled adjacent voxels. Note that, only the unlabeled voxels with non-zero densities are considered; these unlabeled voxels then are inserted into the queue. The third step is repeated until the queue is empty.

We display boundaries between segmented fingers in Fig. 6 by white lines. As a result, the shared regions among finger cores are separated evenly and assigned to the contiguous finger cores by our method. Also, as an illustrative example, the extracted voxels of a finger are presented in Fig. 5d; more complex fingers are displayed in (b) and (c) of both Fig. 7 and Fig. 8. The segmented fingers satisfy the cognition of the earth scientist for individual fingers.

5.2 Construction of Finger Branches

In this section, we describe how to construct finger branches from the 3D finger skeletons. Finger branches are considered as vertical linear structures in the finger skeletons. We create branches from finger skeletons for two reasons. First, branching is critical to domain scientists in understanding

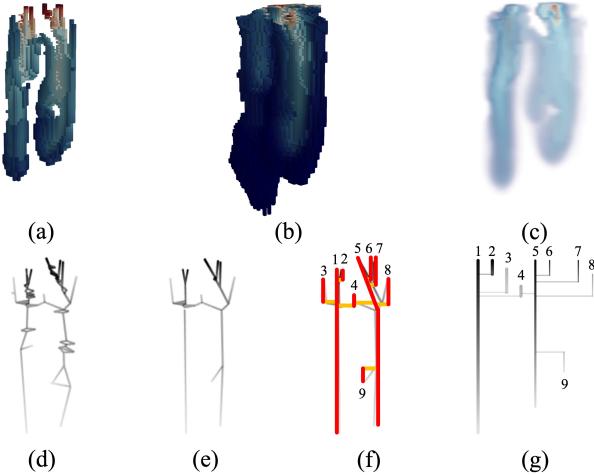


Fig. 7: We illustrate our method by a complex finger. We show (a) the finger core, (b) the complete finger volume, (c) the volume rendering image of (b), (d) the raw finger skeleton, (e) the simplified finger skeleton, (f) the constructed branches of (e) that are represented by red lines and connections between branches that are represented by orange lines, and (g) the linear glyph that is based on (f). The same numbers label corresponding branches.

flow instabilities [14], [15], [16], [17]. Specifically, our earth scientist is more interested in how fingers ramify from top to bottom rather than all the connections within finger skeletons. Second, finger branches as a kind of tree structures can be visualized with minimized occlusion. After building finger branches, we visualize constructed branches by tree-based visualizations to minimize crossings such as Fig. 5g and Fig. 7g.

5.2.1 Spanning Contour Tree for Construction of Finger Branches

We offer a technique which constructs finger branches from finger skeletons. We extract spanning contour trees from the Reeb graph embeddings of fingers to create finger branches, following the suggestion in the works of [18], [19]. Our branch construction method is a heuristic algorithm. Due to the gravity, finger branches stretch vertically. Hence, we assume that each branch grows downward, and capture these vertical structures by our algorithm. The steps of creating finger branches are as follows:

Step 1 Identify vertical structures to be branches from Reeb graphs. Given a graph, we search the longest downward path (i.e., the path with the longest height persistence) to be a new branch by using the breadth-first search. Note that on a downward path, the height of a point should be strictly larger than the height of its following point. If multiple paths in this graph are the longest, we only keep the one with the smallest deviation on the x - y plane. Next, we delete the points and edges of this new branch from the given graph, and repeat Step 1 until the given graph becomes empty. After obtaining all the branches, we insert the longest one into a queue, and mark this one to be enqueued.

Step 2 Build connections between branches. Given a branch from the queue, we identify which unmarked branches have edges with it in the finger skeleton, and record

such edges. If one of the unmarked branches has multiple edges with the given branch, we only keep one of these edges, which is the one that connects to the highest point of the given branch. Finally, we mark the newly identified branches, and insert these branches into the queue.

Fig. 7f shows the branches of a finger and connections between branches created by this algorithm.

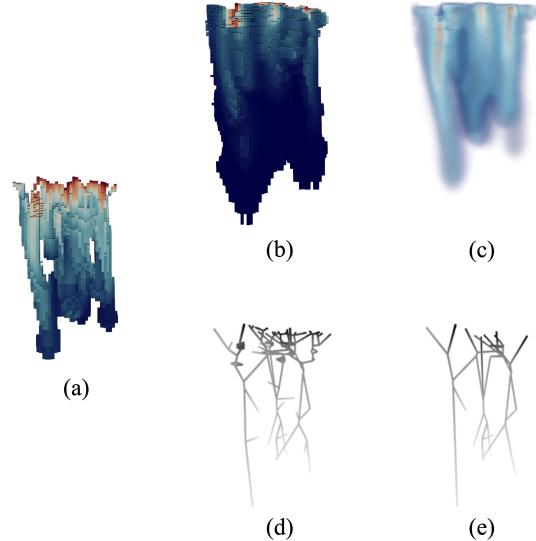


Fig. 8: We display an example of a detected complex finger including (a) the finger core, (b) the complete finger volume, (c) the volume rendering image of (b), (d) the raw finger skeleton, and (e) the simplified finger skeleton.

5.2.2 Simplification of Finger Skeletons by Height Persistence

We simplify the Reeb-graph based finger skeletons. When a finger has a complex structure, small branches may occlude important finger branches and affect the acquisition of essential geometric information (e.g., Fig. 8d). Therefore, a simplified representation of the extracted fingers is essential. The simplification preserves the most relevant geometric information of the fingers and minimizes occlusion such that multiple branches can be analyzed and visualized simultaneously and geometric structure of fingers can be readily understood.

The simplification is based on the geometric characteristic: the height persistence of fingers and their branches. We define *height persistence of a finger* as the height difference between the fingertip and the finger root; similarly, *height persistence of a finger branch* is the height difference between the highest point and deepest point on the branch. Height persistence is an essential attribute of fingers. Due to the gravity, fingers tend to stretch vertically; to analyze the stretching process, we estimate the growing speed of fingers and their branches based on the height. To simplify Reeb graphs, Bauer et al. [40] suggested removing features with persistence smaller than a threshold. Specifically, Carr et al. [41] suggested pruning leaves; in addition to short leaves, Doraiswamy and Natarajan [18] also suggested removing short cycles in Reeb graphs. Hence, to simplify finger skeletons further, we remove short leaf branches (i.e., each of the branches has only one edge to other branches) and certain

short loops. To remove these branches with short height persistence, we remove points on these branches and delete connections related to these points from finger skeletons. We remove the short loops with a simple structure where the loops have short persistence and each of the loops has exact two critical points such as Fig. 1b2. After the removal, some critical points may become passing points in the finger skeletons; we remove these points as well.

A potential concern can be that by only using height persistence, the finger simplification process filters out short but high-density or thick branches, our method supports to add weights for height persistence of branches. Then, weighted height persistence of branches is calculated by average densities or thickness of branches times their height persistence. In this work, we use weighted height persistence for the aforementioned simplification to remedy this concern.

5.3 Volume Overlapping Based Tracking of Fingers and Branches

To track fingers and their branches over time, we need to relate fingers and their branches in consecutive timesteps. Mapping correct features, fingers in this case, in successive timesteps is known as the correspondence problem. Silver and Wang [42] proposed a volume overlapping based method for solving this problem in scalar data. Because fingers usually flow downward and do not move entirely with a large horizontal deviation, recently, researchers have adopted such volume overlapping based methods for tracking fingers. Favelier et al. [12] connected each finger at a timestep only to the finger with the largest volume overlapping at the next timestep to render corresponding fingers in consecutive timesteps by same colors. Aldrich et al. [11] and Lukasczyk et al. [13] relate fingers between adjacent timesteps with any volume overlapping, which offers more connections to track the full temporal evolution of fingers and is used in this paper. To estimate the overlapping volume between fingers in adjacent timesteps, we compare the grid cells of fingers to compute how many cells are shared by fingers in adjacent timesteps. All the fingers who share cells in adjacent timesteps are linked; the numbers of shared cells and densities of these shared cells reflect the strength of connections between linked fingers. Also, the shared cells have positions and can reveal where the fingers overlap. After identifying the correspondence between fingers, we further relate branches of corresponding fingers based on the overlapping of grid cells.

6 VISUALIZATION OF FINGERING

After obtaining the information of viscous and gravitational fingers by applying the aforementioned geometry-driven approach, we create an interactive visual-analytics system to perform spatio-temporal analyses on the geometric structures of the fingers. In our system, we visualize the fingers using spatial and planar glyph representations and present a juxtaposed visualization so that users can compare fingers over space and time. Fig. 9 shows the complete visual-analytics system, which consists of six panels marked as (a)-(f). The depth selection panel (Fig. 9a) allows users to select a depth of interest in the spatial domain. Then, the

density field slice view Fig. 9b displays the density field at the selected depth as a 2D image. The spatial projection panel Fig. 9c presents a high-level spatial view using convex hulls embedded Voronoi cells to indicate the projection of fingers on the slice of the selected depth. Users can also observe different facets of fingers in detail through volume rendering (d) and using a 3D skeleton visualization (e) in Fig. 9. Finally, at the bottom, the geometric-glyph augmented tracking graph Fig. 9f displays the evolution of fingers over time. All the views in our system are interactive and coordinated together to enable coherent visual analyses. Users can select a finger of interest either from the spatial projection view or the geometric-glyph augmented tracking graph for detailed analysis. The selected finger is then highlighted using red color on both the spatial panel and the tracking graph as shown in Fig. 9cf. The 3D volume and skeleton of the selected finger are visualized simultaneously so that users can understand the geometric structures and branching information of the finger in detail. In the following, we describe each of these panels in detail.

6.1 Depth Selection and Density Field Slicing

Density field slices are important for users to study the change of densities inside a finger. To enable this, our system allows users to drag a depth slider in the depth selection panel (Fig. 9a) to select a depth of interest. The density field slice at the selected depth is then extracted and shown in Fig. 9b as a heatmap. In addition, the 3D spatial region below the selected depth is highlighted by gray color in a 3D cube in the depth selection panel. The finger volume and skeleton views are updated to show finger structures under the selected depth only. Initially, selected depth is set to the top of the 3D domain because CO₂ is injected from the top of the aquifer and high-density areas on the top of the 3D domain indicates where CO₂ is concentrated.

6.2 Spatial Projection of Fingers

Before showing individual fingers, a high-level spatial view is required to reveal the relative spatial positions between viscous and gravitational fingers so that users can efficiently see how fingers interact spatially. In this work, we use spatial projection to visualize the relative spatial positions between fingers. We use two methods to project the fingers, including convex hull and Voronoi treemap, as detailed below.

Convex hull We project critical points of finger skeletons onto the *x-y* plane and draw a convex hull to enclose the projection of the critical points of each finger skeleton which helps us to demonstrate the coverage of each finger. By visualizing the convex hulls such as the first row of Fig. 12, we can quickly identify large fingers. This view also supports the selection of a finger of interest to display the volume and skeleton of the selected finger. The convex hull of a selected finger is also superimposed in the density field slice view as shown in Fig. 9b to confirm the existence of the selected finger in the density field. In addition, we project centers of fingers onto the *x-y* plane as points in both the density field slice and spatial projection view; the center of each finger is the center of mass of the finger skeleton. The

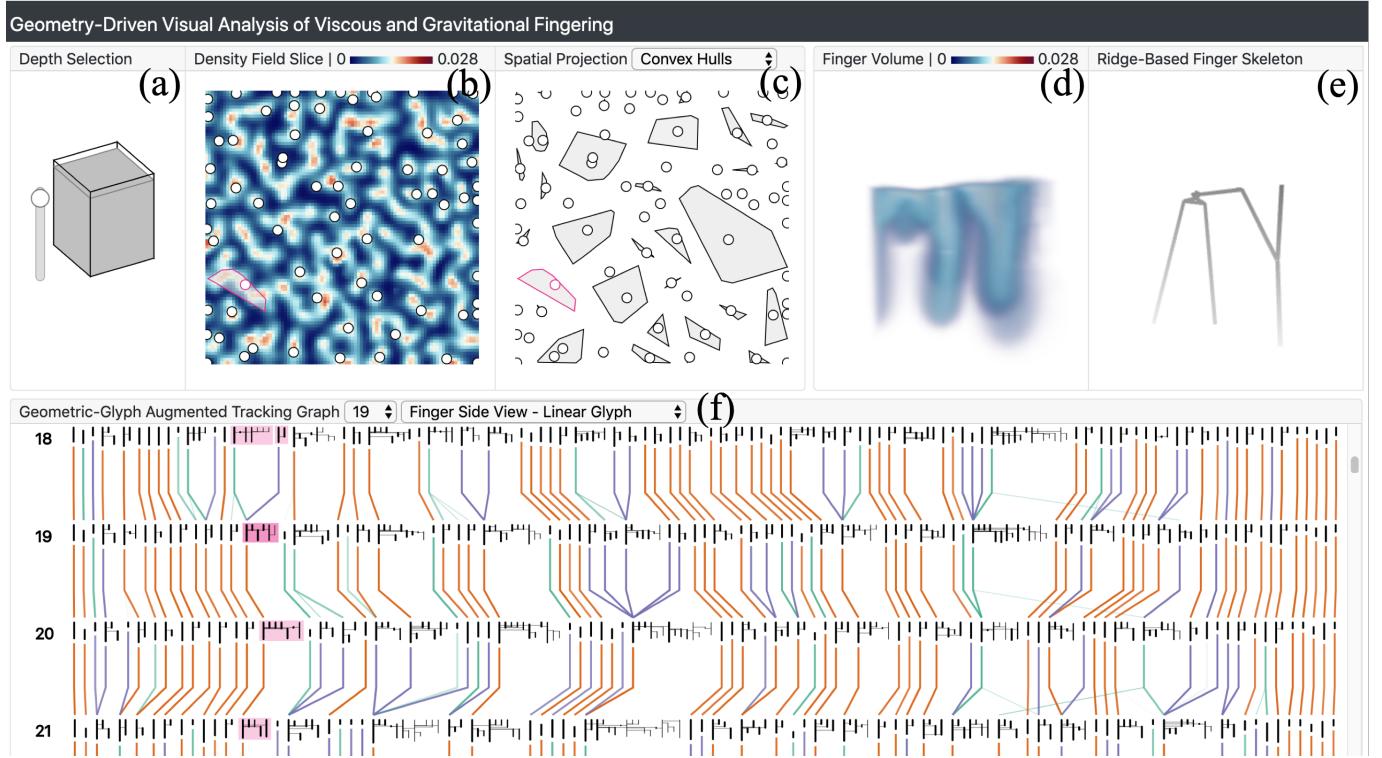


Fig. 9: A geometry-driven visual-analytics system to study viscous and gravitational flow instabilities. (a) Depth slider is used to specify a depth of interest. 3D domain under the specified depth will be shown for analysis. (b) View of a density slice at the specified depth, which also shows projection points of finger-centers onto the x - y -plane. (c) Spatial projection view displays the projection of fingers on the x - y plane of the 3D domain using scattered convex hulls; Voronoi cells that are inside convex hulls show the topological complexity of finger branches. (d) Finger volume view displays density scalar field of a selected finger. (e) Finger skeleton shows the geometric structures of a selected finger in three-dimensional space. (f) Geometry-nested tracking graph visualizes the geometric evolution of fingers over time using various geometry-driven planar glyphs.

spatial projection of these finger centers can illustrate the relative positions between fingers in the spatial domain.

Voronoi treemap Besides visualizing fingers using convex hull, we also use the Voronoi treemap technique [43] to generate Voronoi cells for finger branches, and embed the resulting Voronoi cells in the convex hulls of fingers instead of centroid points to reveal the complexity of finger branches simultaneously such as Fig. 10f and the second row of Fig. 12. Users can select to observe either centroid points for simplicity or Voronoi cells for the summarization of finger branches. Each Voronoi cell corresponds to a finger branch. Dark Voronoi cells (e.g., in Fig. 10f) mean that the corresponding finger branches exist in the deep region of the 3D domain; more precisely, the darkness of Voronoi cells encodes average depth of critical points that are in the corresponding finger branches. Size of a Voronoi cell encodes topological complexity of the corresponding finger branch. We define *topological complexity* of a finger branch by the number of critical points in the skeleton of this finger branch. The intuition behind this definition is that a finger branch with more critical points usually means this finger branch is connected to multiple other branches and thus is more complex in terms of its topological structure. Moreover, while analyzing the fingers, if scientists need to under-emphasize low-density or thin branches, the densities at critical points and the average thickness of branches that

contain the critical points can be used as weights of critical points for calculating the complexity.

6.3 3D Spatial Visualization of Fingers

When a specific finger of interest is selected by the user, we provide 3D volume and skeleton based visualization of the selected finger for further analyses.

6.3.1 Density Field-based Visualization of Fingers

The density field of each finger is visualized using volume rendering to show how fingers appear in the physical domain as shown in Fig. 10bd which allows scientists to interpret the fingers effectively. Moreover, the volume rendering remedies the occlusion on the direct display of voxels (Fig. 8b) with transparency.

To unify the rendering process for fingers with different sizes, we re-sample each extracted finger volume into a 3D regular grid with a fixed resolution (64^3 by default in our application or a user-defined value); then, we apply the standard volume ray casting method to the sampled volume to create volume rendering results for fingers (e.g., Fig. 9d). Users can interactively rotate the volume and zoom in or out to study the structure of the finger from different viewpoints.

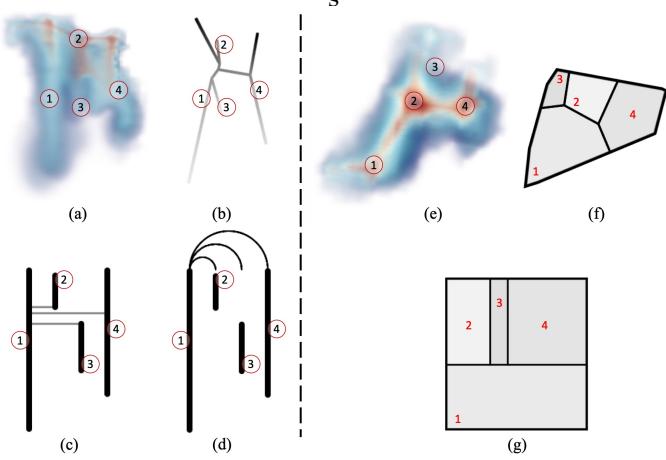


Fig. 10: We showcase glyph designs of a finger and indicate corresponding branches between 3D structures of the finger and the 2D designs of the finger by numbers. Left four views display side views of the finger. (a) shows the side view of the finger volume. (b) is the detected skeleton of the finger. The linear glyph (c) displays connections between branches of the finger. When horizontal connections are too close or even overlap, the alternative design (d), which draws connections between branches as arcs at the top of the design, can better show the tree structure than (c). Right three views display top views of the finger. (e) is the top view of the finger volume. The Voronoi glyph (f) is shown on the spatial projection panel. The rectangular glyph (g) is for contrasting the topological complexity between the finger branches on the tracking graphs.

6.3.2 Ridge-based Geometric Skeleton Visualization of Fingers

To visualize the overall geometric structures of fingers in 3D space, we generate the simplified finger skeletons on the screen using orthogonal projection (e.g., Fig. 10b). To recover the density information of fingers, intensity gradients along lines (e.g., Fig. 10b) are used to indicate changes in the density of finger branches; hence, high-density branches are highlighted, and low-density branches are underemphasized. Users can also rotate and zoom into the finger skeletons to obtain more details of the geometric structures.

6.4 Geometric-Glyph Augmented Tracking Graphs of Fingers

We show the temporal evolution of all the fingers by geometric-glyph augmented tracking graphs as shown in Fig. 9f and Fig. 11. The tracking graphs are targeted at helping domain scientists to identify various evolutionary events of fingers such as finger merging and splitting, and analyze how the evolutionary events happen in detail (e.g., Fig. 14). Each row of the tracking graph displays the extracted fingers of one timestep; the timestamp is labeled at the left of the panel. Note that, we do not present the relatively smaller fingers in this view since the domain expert is primarily interested in tracking the large fingers in the system. Widths of finger glyphs of each row are adjusted according to the topological complexity of finger structures; complex fingers have higher widths for drawing.

Users can switch the style of the tracking graph between the linear glyph (Fig. 9f) and the rectangular glyph (Fig. 11). In addition, users can click on a particular glyph to observe the details of the corresponding finger in the volume rendering and skeleton views.

Below we describe the tracking graph in more details including the generation of geometric glyphs for showing structures of fingers, the color encoding of links for evolutionary events of fingers, the minimization of link crossings for reduction of visual clutter, and the interactions for exploration of temporal events.

6.4.1 Visualization of Finger Side View by Tree Drawing Method

To display the evolution of geometric structures of fingers, we enhance tracking graphs by nesting geometry-driven glyphs on the tracking graphs. To draw the side view of fingers, according to the requirements of domain experts, we have to draw finger branches and their connections in a plane with minimized occlusion and also preserve branch heights. Two possible solutions are discussed in the following.

Projection. We can project finger skeletons onto a plane. However, the traditional orthogonal projection suffers from edge crossing problems. To remedy the edge crossing for tree-like structures, the method [44] produces radial planar embeddings. However, fingers are vertical objects, and radial planar embeddings of fingers can lose the height information of the fingers. Although this method [44] is good at preserving the curvature of 3D objects, the curvature is not an essential quantity for fingers.

Tree drawing. Secondly, we can draw fingers on a plane with a linear glyph. Heine et al. [19] proposed a graph-drawing method to draw branches of contour trees in a plane with minimized edge crossings. This method [19] represents branches by vertical lines and shows links between branches by horizontal lines. Heights of branches and connections between branches are shown clearly in the results of this method. Thus, we augment the tree-drawing method of Heine et al. [19] to display the side view of fingers such as Fig. 10c. Note that, although trees are planar graphs, crossings of tree edges are inevitable in this design, since y -axis is used to encode the height of branches, and only the x -axis is free to arrange tree branches. We draw the longest branch of a finger at the leftmost of the glyph so that we can locate the principal branch quickly. Also, we remove short branches to keep simplicity. Results of the tree-drawing method are simplified forms of fingers and can display branch connections and heights effectively. Users can choose to encode change of densities along branches to the gradient darkness of vertical lines such as Fig. 5c. When hovering a linear glyph on the tracking graph, connections between branches of this finger become arcs hanging at the top of the glyph to reduce visual clutter (i.e., Fig. 10d).

6.4.2 Visualization of Finger Top View by Spatially Ordered Treemap

To draw the top view of fingers, we need to display quantitative statistical attributes and relative positions of fingers. The treemap technique was developed by Johnson and Shneiderman [45], who mapped elements onto a

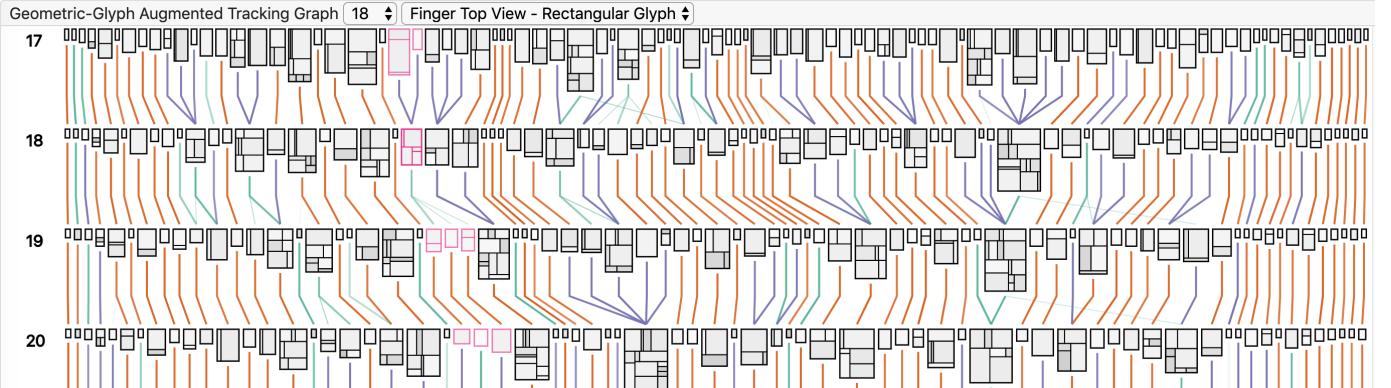


Fig. 11: The tracking graph displays the temporal evolution of fingers represented as rectangular glyphs. A finger associated with merging and splitting events at timestep 18 is highlighted by red-violet and analyzed in Fig. 14 further.

rectangular region in a space-filling manner and displayed quantity values of elements effectively. Spatially ordered treemap method [20] proposed by Wood and Dykes extends squarified treemaps [46] to incorporate spatial information of elements.

We use the spatially ordered treemap method to generate rectangular glyphs for finger branches as shown in Fig. 10g. Each branch is represented by a rectangle in the glyphs. Areas of rectangles correspond to the topological complexity of branches (defined in Sect. 6.2). The branches are arranged based on both the spatial proximity and the balance of aspect ratio. Moreover, to compare fingers in the tracking graph Fig. 11 and assign more space for complex fingers, we encode the size of each rectangular glyph by the topological complexity of fingers. The *topological complexity* of a finger is defined by the number of critical points in its finger skeleton; a finger has more critical points usually means this finger has more branches and hence is more complex in its topological structure. The resulting rectangular glyphs are relatively stable over time.

6.4.3 Link Encoding for Evolutionary Events of Fingers

We link fingers which share cells between adjacent timesteps. **Link styles.** When time varies, fingers may grow, merge with other fingers, or split into multiple fingers. To indicate these three types of the evolution of fingers, we use three different hues of links following qualitative color advice of ColorBrewer [47].

Brown Link The fingers with brown links grow with minor changes in its branches in consecutive timesteps. At least seventy-five percent volume of the finger is preserved in the connected finger at the subsequent timestep.

Blue Link The fingers with blue links represent the case when multiple fingers merge into one finger at a consecutive timestep. The finger in the consecutive timestep incorporates seventy-five percent volume of each of the fingers at the previous timestep.

Green Link The fingers with blue links split into multiple fingers at the subsequent timestep, and none of the fingers at the subsequent timestep have seventy-five percent volume of the finger.

Moreover, we use the opacity of links to encodes link weights; the weight of each link is the size of the overlapping volume between fingers. If fingers have weak connec-

tions, their links are almost transparent so that visual clutter is reduced.

6.4.4 Iterative Minimization of Link Crossings

We reduce link intersections to alleviate visual clutter in the tracking graph further. Since links have weights and connect fingers between adjacent timesteps, the reduction of link crossings in our application is a graph drawing problem: edge crossing minimization in weighted bipartite graphs. Çakiroğlu et al. [48] enhanced and tested five heuristic methods for this graph drawing problem and concluded that W-GRE [48], [49] and W-PM [48], [50] methods produce the best results. Hence, we utilize W-GRE method in our application to minimize weighted link crossings.

We apply the W-GRE method to arrange fingers on each row iteratively until obtaining a good result.

Step 1 The fingers on the first row are initialized by the ascending order of their x coordinates.

Step 2 To order fingers from the second row to the last row, we minimize the crossings of the links between each row and its previous row by using the W-GRE method.

Step 3 To reorder fingers from the last but one row to the first row, we minimize the crossings of the links between each row and its following row by using the W-GRE method.

Step 4 Repeat Step 2 and Step 3 to reorder fingers of each row until reaching the convergence of the finger order. Usually, repeating two times can obtain a good result.

6.4.5 Interactive Tracking for Exploring Finger Branches and Volume

For scientists to track fingers effectively, we offer three interactions. For users to follow the evolution of fingers, after users select a finger of interest, the selected finger and the other related fingers are highlighted in the tracking graphs by coloring the backgrounds of the fingers as shown in Fig. 9f or the strokes in Fig. 11.

The earth scientist, involved in this work, also requires designs to track branches. If a complex finger separates into multiple smaller fingers over time, it is essential to know each branch of the complex finger comes to which smaller finger entity afterwards; also, if multiple fingers fuse into a complex finger, it is important to understand the correspondence between the individual fingers and the

branches that they got merged to. Through our geometric-glyph augmented tracking graph, we can identify corresponding branches between linked fingers interactively. When hovering over a link between two connected fingers, having identified relevant branches in Sect. 5.3, representations of these corresponding branches in glyphs of the two connected fingers will be highlighted by red-violet color as shown in Fig. 14.

Furthermore, the earth scientist is interested in tracking the volume of a finger. Note that each link is associated with an overlapping volume between connected fingers. Users can click on a finger of interest first and click on associated links to observe overlapping volume between this finger of interest and one of its connected fingers in the finger volume view, which is illustrated in Sect. 7.3 and Fig. 14 in detail.

7 CASE STUDY AND EXPERT FEEDBACK

In this section, we employ our geometry-driven visual-analytics system to explore the phenomena of viscous and gravitational fingering and perform domain-specific tasks. After we have built the visual-analytics system, we discussed our work with the earth scientist. The earth scientist and his graduate student both thought that this work provides a new perspective to analyze viscous and gravitational fingering. They acknowledged that the extraction of fingers in this way is effective and that the minimal occlusion leads to a clear representation of the fingers. Our earth scientists also gave us valuable suggestions about our system. Below we discuss several use cases that were studied using our system.

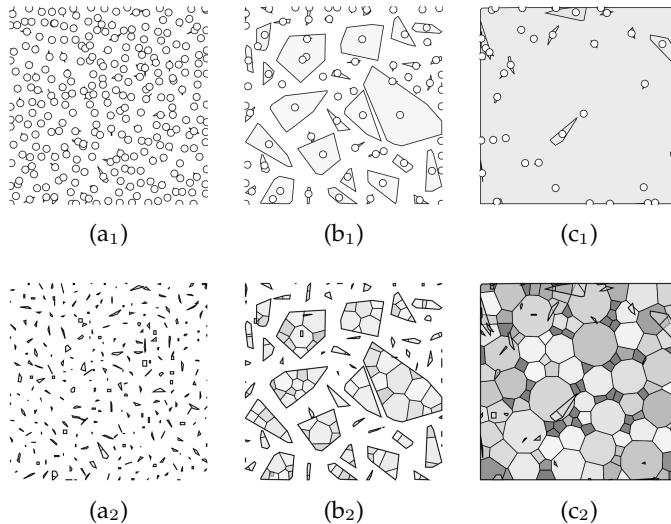


Fig. 12: The spatial projection of fingers for three phases of the evolution of fingers: (a) the onset phase, (b) the growth phase, and (c) the termination phase. (a₁), (b₁), and (c₁) show the convex hull representations. (a₂), (b₂), and (c₂) display the Voronoi glyphs.

7.1 Case 1: Recognition of Evolutionary Phases

According to the earth scientist, the evolution of the instability may have (at least) three important phases: onset, growth, and termination, which have also been studied in

previous works [2], [12], [13]. The earth scientists identified the three phases based on the spatial projection panel. In the onset phase, the instability is triggered, and the displacement front breaks up in many small-scale fingers, which can be identified in Fig. 12a. After the onset phase, the dense fingers grow non-linearly and penetrate the underlying lighter fluid. In Fig. 12b, many medium-sized fingers form in this growth phase. These medium-sized fingers grow, merge, and/or split over time. Eventually, multiple fingers merge to form a few ‘mega-plumes’ that span most of the reservoir, as in Fig. 12c. Once those fingers reach the bottom of the domain, the instability shuts down or terminates.

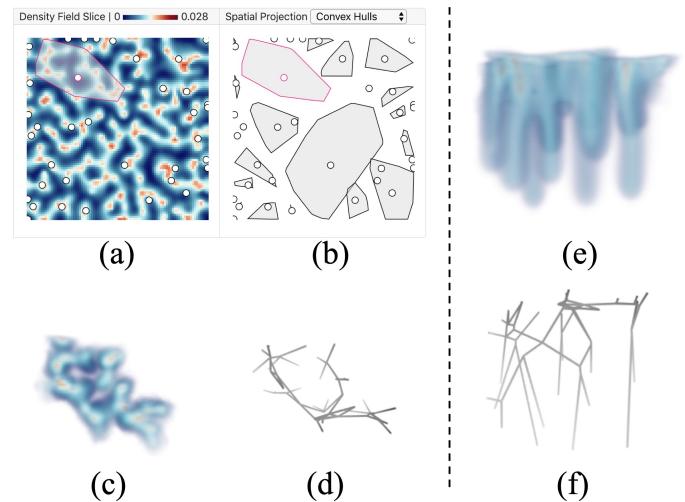


Fig. 13: The earth scientists identified a finger of interest from (b) the spatial projection panel. Then, they observed the high-density hexagons under the convex hull of the finger in (a) the density field slice view. They further confirmed the correspondence between (c) the top view of the finger volume and the hexagons in (a). Also, they found the finger skeleton (d) preserved the geometric structure of (c). Next, they looked at the side view of the finger volume (e) but felt difficult due to the occlusion of finger branches. They then interacted with the 3D skeleton (f) and can obtain how these branches form in space over time.

7.2 Case 2: Identification of Fingers in 3D Space

Because the previous methods do not capture the geometric structures of fingers (e.g., finger skeletons) on purpose, our earth scientist mentioned the difficulty in tracing density patterns of fingers in 3D space, due to occlusion. Specifically, there are three problems. The first is to identify where are finger branches in space. Second, the problem is to understand that how the CO₂ volume flows in space and time and how the hexagonal sheets in the top (e.g., Fig. 13ab) split and develop into columnar fingers (e.g., Fig. 13ef) below. Third, given hexagonal cells (e.g., Fig. 13cd), it is important to know that whether fingers tend to form along with the boundaries of hexagonal cells or form below centers of the hexagonal cells. By using our system, the earth scientists started solving the three problems.

To identify where are finger branches, the scientists first looked at the density field slice view and spatial projection view. The earth scientists thought projection points added on the density field slice view, Fig. 13a, is an effective

addition to traditional density-slice-views and allow them to identify finger locations on any slice with the corresponding hexagonal features at the top. Furthermore, from the spatial projection panel, Fig. 13b, the earth scientists found two fingers that occupy a large space. They clicked on one of the two for analysis; the convex hull of the finger was superimposed on the density field slice view, Fig. 13a. They focused on the high-density hexagons under the convex hull in the slice view, and adjusted the z -value slider in Fig. 9a to understand the change of densities of slices along the z coordinate. Moreover, the finger volume (Fig. 13ce) and skeleton (Fig. 13df) views were displayed. They verified the correspondence between the hexagons under the convex hull in Fig. 13a and the top view of the volume, Fig. 13c. Next, they confirmed the correspondence between the finger volume and skeleton. The skeleton Fig. 13d captured the structure of the volume Fig. 13c. Afterwards, they rotated finger volume and skeleton to see side views of the finger. The finger branches occluded severely in Fig. 13e. However, when interacting with the skeleton Fig. 13f, they could quickly acquire the geometric structures of the finger and identify finger branches in space.

To understand how CO_2 volume flows in space and how fingers form in time, since the finger results from CO_2 flows over time, the finger skeleton at a timestep (Fig. 13f) approximates the flowing track of CO_2 over time; hence, the earth scientists could answer whether the high-density hexagons form this finger over time, and understand how these branches form.

For the third problem, based on the finger skeletons (e.g., Fig. 13f), the experts found that the fingers usually form along the boundaries of hexagonal cells instead of forming below centers of hexagonal cells.

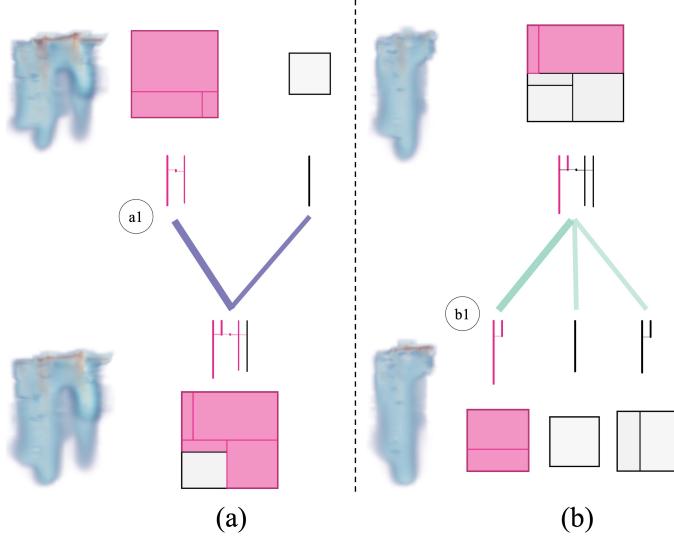


Fig. 14: After evolving one timestep, fingers may (a) merge into a complex finger or a complex finger may (b) split into more fingers. To track branches, users can hover over a link; the link becomes thick and associated branches are highlighted by red-violet color. To track the volume of fingers, users can click on a link; then, the overlapping volume between linked fingers is shown to be compared.

7.3 Case 3: Exploration of Finger Evolution

In this final case study, the earth scientists identified the growth, merging, and splitting events of fingers over time from our geometric-glyph augmented tracking graph. First, we illustrated the encodings of our designs to the earth scientists. After we explained the three colors used to represent the growing, merging, and splitting of the fingers, they appreciated how the tracking graph could provide insights into the temporal evolution of the fingers. We next explained the two kinds of finger glyphs to the earth scientists. Comparing with the tracking graphs in the previous approaches which only use dots to represent fingers, our geometric-glyph augmented tracking graphs can offer more details and facets of fingers when the earth scientists interact with the tracking graphs. With respect to the linear glyphs (Fig. 10c), at first glance, they were confused with the encoding of horizontal lines. However, after we explained that the horizontal lines are used to represent connections between vertical branches, they were able to understand the structures and thought that this design is intuitive. They thought drawing fingers in a plane would cause distortion, but found that the branches and their connections are shown clearly, and it is easy to count the number of vertical branches. In regards to the rectangular glyph, the expert was able to comprehend and quickly count the number of complex branches. Also, they suggested highlighting the correspondence between the rectangles and the branches in the volume so that they can understand the structures of fingers from the glyphs well.

In the following, we illustrate how the earth scientists analyzed the evolution of fingers and branches by our system. Comparing with the previous papers which do not construct branches of fingers explicitly, our methods can track finger branches efficiently. First, they found a finger of interest with merging and splitting events over time, which is highlighted by red-violet in Fig. 11. Concentrating on this finger, they hovered on the left link above the merged finger in Fig. 14a and the leftmost link below the finger in Fig. 14b to track corresponding branches, which were highlighted by red-violet. They found that the corresponding linear glyphs and rectangular glyphs are consistent. To track the volume, they first clicked on the upper-left finger (a1) of Fig. 14a to look at its volume; then, they clicked on the merged finger at the next timestep and clicked on the link to (a1) to observe the partial volume of the merged finger that comes from (a1). The two images of the volume are shown in Fig. 14a and are consistent. Similarly, they tracked the overlapping volume of fingers in Fig. 14b and observed the growth of corresponding branches. After using our system, the earth scientists thought our tracking graphs are valuable to identify when and where the fingers merge and split over time.

8 DISCUSSION

Sensitivity of identified finger cores to the stability of ridge detection. When data has low quality, for example, data has noise or low discretization, the approximation of derivatives of scalar fields usually has high errors. Since the ridge-voxel detection is based on the approximated derivatives, the results of the extraction of finger cores usually also

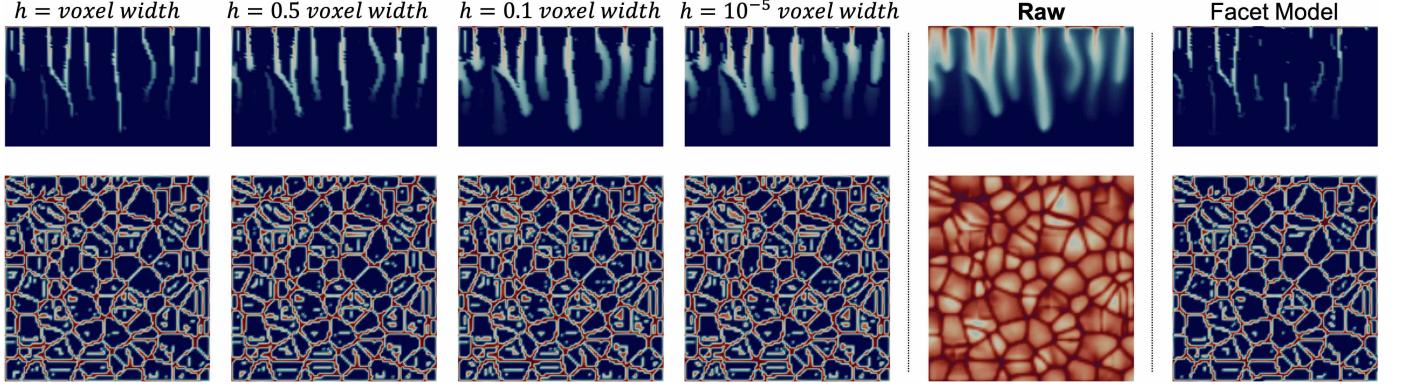


Fig. 15: We display results of Method One with varying h values and Method Two for the fingering data. The first row is for the side view, and the second row is for the top view.

have high errors. Also, if the density fields are not smooth enough, the resulting ridge lines are likely broken; hence, it will be harder to extract continuous finger cores.

Limitations of the height value selection for the top layer. Following the previous papers [5], [11], [13], in our work, we select a fixed height value as the separation between the top layer and the bottom layer of the whole domain. Physically, the top layer initially grows diffusively in time, i.e., as the square root of time, which is much slower than the convective time scale over which the fingers grow. Once the instability is triggered, this diffusive boundary layer thins. Since the top layer is varying in nature, one limitation of using a fixed height value for the top layer is that the fixed height value for the top layer may not be suitable for all conditions. However, given the domain ranges from 40 to 0, we find that, for our selected fixed height value 38, the fingers are separated well throughout all timesteps of our dataset. For other datasets, our system provides visualizations for scientists to identify a suitable value for their datasets. The other limitation is that the number of fingers is sensitive to the height value for cutting. However, the influence is remedied by interactions with the spatial projection view. For example, if the height value is set low, a finger with multiple branches may become multiple individual fingers; in the projection view, these individual fingers are close. By interacting with the slice view, we visualize the density fields among these close individual fingers and identify the connections between them to determine whether the height value is set too low or not.

Scalability of geometric-glyph augmented tracking graph. Given a screen resolution of 2880×1800 , our geometric-glyph augmented tracking graphs support displaying around one hundred fingers and hundreds of branches for each timestep, and around three to four timesteps when placed in our visual-analytics system. When the number of fingers of a timestep becomes large, it will be hard to arrange all the fingers of a timestep in the same row by using our tested screen resolution. To remedy this scalability issue, a larger screen, more simplification of representations of fingers, and allowing multi-level exploration of fingers may be helpful.

Future works. From the interactions with our earth scientists, we identify future works. The 3D cube in the

depth selection panel Fig. 9a can be enhanced to display the selected finger in 3D space by wired structures. Also, we may add an iso-contour view to observe and track the iso-contours of the scalar fields. Additionally, a line chart to display the number of merging and splitting events over time is essential for the temporal analysis of fingers and associated mixing and spreading processes of, e.g., CO_2 or contaminants. Furthermore, our collaboration with the earth scientists is ongoing to use the visualization and analysis tools developed in this work to study the scaling behavior associated with pattern formation in flow instabilities; especially, the topological measurements, that are offered by our methods, including the number of branches and the number of critical points can be studied for discovering new scaling laws.

9 CONCLUSION

In this paper, we present a novel geometry-driven approach with a finger core detection, a finger skeletonization approach, and a finger branch extraction method for the detection of viscous and gravitational fingering. An interactive visual-analytics system with a geometric-glyph augmented tracking graph is established for domain scientists to track the geometric growth, merging, and splitting of fingers over time in detail. The earth scientists recognized the value of our work and thought that this could reduce their analysis workload both in space and time significantly.

APPENDIX A DERIVATIVE APPROXIMATION FOR RIDGE VOXEL GUIDED EXTRACTION OF FINGER CORES

Method One: Assume the raw density function is $g(x, y, z)$, we approximate its derivatives as follows. In Fig. 16a, the blue points of the right figure represent midpoints of voxels. We assume density values at blue points are values of voxels exactly; densities at other points are estimated by trilinear interpolation of densities of blue points. We take the estimation of the partial derivative for x coordinate at the red-circled point as an example. We use the adjacent two orange points, which are h units away from the red-circled midpoint; the density values of the two orange points are $g(x + h, y, z)$ and $g(x - h, y, z)$ respectively. Then, we use

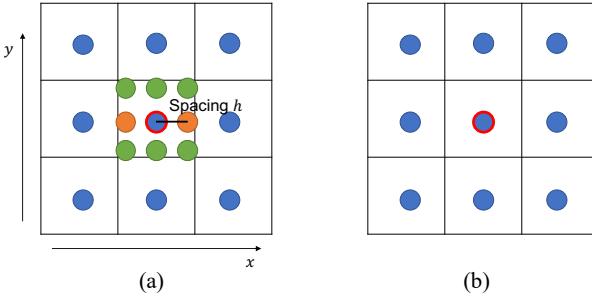


Fig. 16: The two illustration figures (a) (for Method One) and (b) (for Method Two) are used to help elucidate the two derivative approximation methods in App. A.

one of the finite difference formulas, the central difference formula, to estimate the derivatives:

$$g_x(x, y, z) \approx \frac{g(x + h, y, z) - g(x - h, y, z)}{2h} \quad (6)$$

When h is equal to the width of a voxel, the orange points overlap with two adjacent blue points of the red-circled point; hence we use densities of the two blue points directly to estimate the derivative; otherwise, we use the tri-linear interpolation to estimate the densities at orange points to approximate the derivative. Derivatives for other coordinates or directions are calculated similarly by using their corresponding central difference formulas. The results with varying values of h for extracting fingers are shown in the left side of Fig 15. Generally, when h is large, the extracted features are thin. When h is small, the detected features are thick and preserve more branching connections.

Method Two: We also test the facet model [51] which approximates parameters of Taylor polynomials locally by using least squares method. Assume the raw density function is $g(x, y, z)$, and we use the Taylor polynomial (Equation 3) to approximate the function g near the red-circled point locally. The unknown parameters are the coefficients in $J(\mathbf{p}_0)$ and $H(\mathbf{p}_0)$, i.e., the derivatives. In Fig. 16b, given the densities of the blue points near the red-circled point, we use the least squares method to optimize the following function to acquire the unknown parameters.

$$\text{minimize}_{\mathbf{p}} \sum_{\mathbf{p}} (f(\mathbf{p}) - g(\mathbf{p}))^2 \quad (7)$$

The results of this method for the fingering data are displayed in the rightmost column of Fig 15.

Results on synthetic data. Furthermore, we test both of the aforementioned methods on synthetic data. The volume rendering of the synthetic data is shown in Fig. 18; the columnar and spiral shapes have higher densities at their central regions. The results for both methods on this data are shown in Fig 17. We find that the facet model (Method Two) produces thin features that are similar to the Method One with h that equals to the width of voxel; these results are ridge lines. As we decrease h for Method One, the detected features become thicker. When $h = 10^{-5}$ voxel width, each of the detected feature is almost each individual spiral column. Thus, we can separate spiral columns in the synthetic data and extract each individual spiral column based on the results when $h = 10^{-5}$ voxel width by using

the same watershed-based segmentation method when we separate fingers.

Summary. From this study, we find that the finger structures produced by Method One in Fig 15 are more continuous than the fingers detected by Method Two, the facet model. Hence, we use Method One to estimate the derivatives for our fingering data. Moreover, the first method is more flexible since domain scientists can control the values of h to acquire fingers by trading off thickness against more branching connections of structures. The scientists can vary this parameter and from their experience can find a suitable value for h . In our paper, by default for h , we use a small value of h (i.e., 10^{-5}) to extract fingers; since when h is small, the branching connections of structures are preserved well, features are also separated well, and are close to the complete volume of fingers. We further use Reeb-graph based skeletonization to acquire thin skeletons.

ACKNOWLEDGMENTS

This work was supported in part by UT-Battelle LLC 4000159557, Los Alamos National Laboratory Contract 471415, and NSF grant SBE-1738502. We also thank Yujia Wang and Dr. Junpeng Wang to provide design suggestions for the geometric-glyph augmented tracking graph.

REFERENCES

- [1] J. Moortgat, "Viscous and gravitational fingering in multiphase compositional and compressible flow," *Advances in Water Resources*, vol. 89, pp. 53–66, 2016.
- [2] M. A. Amooie, M. R. Soltanian, and J. Moortgat, "Solutal convection in porous media: Comparison between boundary conditions of constant concentration and constant flux," *Physical Review E*, vol. 98, no. 3, p. 033118, 2018.
- [3] G. M. Homsy, "Viscous fingering in porous media," *Annual review of fluid mechanics*, vol. 19, no. 1, pp. 271–311, 1987.
- [4] M. A. Amooie, M. R. Soltanian, and J. Moortgat, "Hydrothermodynamic mixing of fluids across phases in porous media," *Geophysical Research Letters*, vol. 44, no. 8, pp. 3624–3634, 2017.
- [5] T. Luciani, A. Burks, C. Sugiyama, J. Komperda, and G. E. Marai, "Details-first, show context, overview last: Supporting exploration of viscous fingers in large-scale ensemble simulations," *IEEE Transactions on Visualization and Computer Graphics*, 2018.
- [6] A. Skauge, P. A. Ormehaug, T. Gurholt, B. Vik, I. Bondino, G. Hamon *et al.*, "2-d visualisation of unstable waterflood and polymer flood for displacement of heavy oil," in *Proc. SPE Improved Oil Recovery Symposium*, 2012.
- [7] M. R. Soltanian, M. A. Amooie, Z. Dai, D. Cole, and J. Moortgat, "Critical dynamics of gravito-conective mixing in geological carbon sequestration," *Scientific Reports*, vol. 6, p. 35921, 2016.
- [8] M. A. Amooie, M. R. Soltanian, F. Xiong, Z. Dai, and J. Moortgat, "Mixing and spreading of multiphase fluids in heterogeneous bimodal porous media," *Geomechanics and Geophysics for Geo-Energy and Geo-Resources*, vol. 3, no. 3, pp. 225–244, 2017.
- [9] M. R. Soltanian, M. A. Amooie, N. Gershenson, Z. Dai, R. Ritzi, F. Xiong, D. Cole, and J. Moortgat, "Dissolution trapping of carbon dioxide in heterogeneous aquifers," *Environmental Science & Technology*, vol. 51, no. 13, pp. 7732–7741, 2017.
- [10] X. Fu, L. Cueto-Felgueroso, and R. Juanes, "Pattern formation and coarsening dynamics in three-dimensional convective mixing in porous media," *Phil. Trans. R. Soc. A*, vol. 371, no. 2004, p. 20120355, 2013.
- [11] G. Aldrich, J. Lukasczyk, M. Steptoe, R. Maciejewski, H. Leitte, and B. Hamann, "Viscous fingers: A topological visual analytics approach," *IEEE Scientific Visualization Contest*, vol. 1, no. 2, p. 4, 2016.
- [12] G. Favelier, C. Gueunet, and J. Tierny, "Visualizing ensembles of viscous fingers," in *IEEE Scientific Visualization Contest*, 2016.

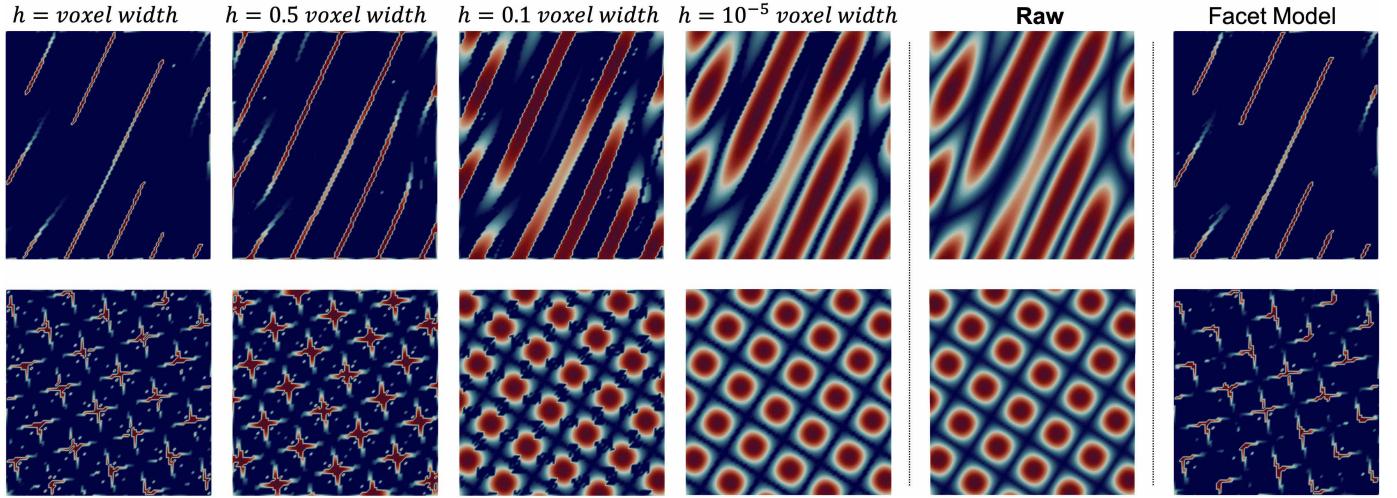


Fig. 17: We display results of Method One with varying h values and Method Two for synthetic data. The first row is for the side view of Fig 18, and the second row is for the top view.

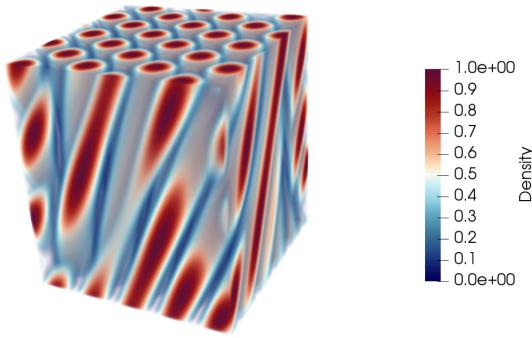


Fig. 18: This figure shows the volume rendering image of the synthetic data, which is consisting of columnar and spiral shapes with high-density central regions.

- [13] J. Lukasczyk, G. Aldrich, M. Stepto, 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*, vol. 869, 2017, pp. 9–19.
- [14] A. Buka and P. Palffy-Muhoray, "Stability of viscous fingering patterns in liquid crystals," *Physical Review A*, vol. 36, no. 3, p. 1527, 1987.
- [15] A. Buka, P. Palffy-Muhoray, and Z. Racz, "Viscous fingering in liquid crystals," *Physical Review A*, vol. 36, no. 8, p. 3984, 1987.
- [16] E. L. Hinrichsen, K. Maloy, J. Feder, and T. Jossang, "Self-similarity and structure of dla and viscous fingering clusters," *Journal of Physics A: Mathematical and General*, vol. 22, no. 7, p. L271, 1989.
- [17] S. Li, J. S. Lowengrub, J. Fontana, and P. Palffy-Muhoray, "Control of viscous fingering patterns in a radial hele-shaw cell," *Physical review letters*, vol. 102, no. 17, p. 174501, 2009.
- [18] H. Doraiswamy and V. Natarajan, "Output-sensitive construction of reeb graphs," *IEEE transactions on visualization and computer graphics*, vol. 18, no. 1, pp. 146–159, 2012.
- [19] C. Heine, D. Schneider, H. Carr, and G. Scheuermann, "Drawing contour trees in the plane," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 11, pp. 1599–1611, 2011.
- [20] J. Wood and J. Dykes, "Spatially ordered treemaps," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, 2008.
- [21] C. Steger, "An unbiased detector of curvilinear structures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 2, pp. 113–125, 1998.
- [22] F. Lazarus and A. Verroust, "Level set diagrams of polyhedral objects," in *Proc. ACM Symposium on Solid Modeling and Applications*, 1999, pp. 130–140.
- [23] U. Ayachit, "The paraview guide: a parallel visualization application," 2015.
- [24] H. Childs, E. Brugger, B. Whitlock, J. Meredith, S. Ahern, D. Pugmire, K. Biagis, M. Miller, C. Harrison, G. H. Weber, H. Krishnan, T. Fogal, A. Sanderson, C. Garth, E. W. Bethel, D. Camp, O. Rübel, M. Durant, J. M. Favre, and P. Navrátil, "VisIt: An End-User Tool For Visualizing and Analyzing Very Large Data," in *High Performance Visualization—Enabling Extreme-Scale Scientific Insight*, Oct 2012, pp. 357–372.
- [25] "Tecplot," March 2018, <https://www.tecplot.com/>.
- [26] P. De Anna, M. Dentz, A. Tartakovsky, and T. Le Borgne, "The filamentary structure of mixing fronts and its control on reaction kinetics in porous media flows," *Geophysical Research Letters*, vol. 41, no. 13, pp. 4586–4593, 2014.
- [27] J. Lukasczyk, G. Weber, R. Maciejewski, C. Garth, and H. Leitte, "Nested tracking graphs," *Computer Graphics Forum*, vol. 36, no. 3, pp. 12–22, 2017.
- [28] M. de Saint-Venant, "Surfaces à plus grande pente constituées sur des lignes courbes," *Bulletin de la Société Philomathématique de Paris*, pp. 24–30, 1852.
- [29] T. Lindeberg, "Edge detection and ridge detection with automatic scale selection," *International Journal of Computer Vision*, vol. 30, no. 2, pp. 117–156, 1998.
- [30] J. D. Furst and S. M. Pizer, "Marching ridges," in *SIP*, 2001, pp. 22–26.
- [31] R. Peikert and F. Sadlo, "Height ridge computation and filtering for visualization," in *Proc. IEEE Pacific Symposium on Visualization*, 2008, pp. 119–126.
- [32] M. Kern, T. Hewson, F. Sadlo, R. Westermann, and M. Rautenkhaus, "Robust detection and visualization of jet-stream core lines in atmospheric flow," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 893–902, 2018.
- [33] N. D. Cornea, D. Silver, and P. Min, "Curve-skeleton properties, applications, and algorithms," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 3, pp. 0530–548, 2007.
- [34] S. Biasotti, D. Giorgi, M. Spagnuolo, and B. Falcidieno, "Reeb graphs for shape analysis and applications," *Theoretical Computer Science*, vol. 392, no. 1–3, pp. 5–22, 2008.
- [35] M. Van Kreveld, R. van Oostrum, C. Bajaj, V. Pascucci, and D. Schikore, "Contour trees and small seed sets for isosurface traversal," in *Proc. Annual Symposium on Computational Geometry*, 1997, pp. 212–220.
- [36] J. Tierny, G. Favelier, J. A. Levine, C. Gueunet, and M. Michaux, "The topology toolkit," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 832–842, 2018.
- [37] J. Damon, "Properties of ridges and cores for two-dimensional images," *Journal of Mathematical Imaging and Vision*, vol. 10, no. 2, pp. 163–174, 1999.
- [38] D. Eberly, *Ridges in image and data analysis*. Springer Science & Business Media, 2012, vol. 7.

- [39] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 6, pp. 583–598, 1991.
- [40] U. Bauer, X. Ge, and Y. Wang, "Measuring distance between reeb graphs," in *Proc. Annual Symposium on Computational Geometry*, 2014, p. 464.
- [41] H. Carr, J. Snoeyink, and M. van de Panne, "Flexible isosurfaces: Simplifying and displaying scalar topology using the contour tree," *Computational Geometry*, vol. 43, no. 1, pp. 42–58, 2010.
- [42] D. Silver and X. Wang, "Volume tracking," in *Proc. Visualization*, 1996, pp. 157–ff.
- [43] M. Balzer and O. Deussen, "Voronoi treemaps," in *Proc. IEEE Symposium on Information Visualization*, 2005, pp. 49–56.
- [44] J. Marino and A. Kaufman, "Planar visualization of treelike structures," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 906–915, 2016.
- [45] B. Johnson and B. Shneiderman, "Tree-maps: A space-filling approach to the visualization of hierarchical information structures," in *Proc. Visualization*, 1991, pp. 284–291.
- [46] M. Bruls, K. Huizing, and J. J. Van Wijk, "Squarified treemaps," in *Data visualization*, 2000, pp. 33–42.
- [47] C. A. Brewer, "Colorbrewer: Color advice for maps," September 2018, <http://www.ColorBrewer.org>.
- [48] O. A. Çakiroğlu, C. Erten, Ö. Karataş, and M. Sözdinler, "Crossing minimization in weighted bipartite graphs," in *International Workshop on Experimental and Efficient Algorithms*, 2007, pp. 122–135.
- [49] A. Yamaguchi and A. Sugimoto, "An approximation algorithm for the two-layered graph drawing problem," in *International Computing and Combinatorics Conference*, 1999, pp. 81–91.
- [50] K. Sugiyama, S. Tagawa, and M. Toda, "Methods for visual understanding of hierarchical system structures," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, no. 2, pp. 109–125, 1981.
- [51] R. M. Haralick, "Ridges and valleys on digital images," *Computer Vision, Graphics, and Image Processing*, vol. 22, no. 1, pp. 28–38, 1983.



Jiayi Xu is a Ph.D. student in Computer Science and Engineering at the Ohio State University. He received his B.E. degree with Chu Kochen Honors from College of Computer Science and Technology at Zhejiang University in 2014. His research interests include graph visualization and scientific feature tracking.



Soumya Dutta is a staff scientist in Data Science at Scale team at Los Alamos National Laboratory. He received his B.Tech. degree in Electronics and Communication Engineering from West Bengal University of Technology in August 2009, and M.S. and the Ph.D. degree in Computer Science and Engineering from the Ohio State University in May 2017 and May 2018 respectively. His research interests are statistical data summarization and analysis; in situ data analysis, reduction, and feature exploration; uncertainty analysis; and time-varying, multivariate data exploration. He received honorable mention award at IEEE Visualization 2016, best paper award at ISAV 2018, best paper award at TopoInVis 2019.

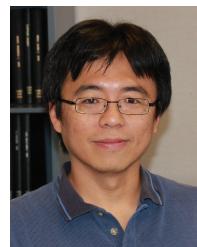


Wenbin He is a Ph.D. student in computer science and engineering at the Ohio State University. He received his B.S. degree from the Department of Software Engineering at Beijing Institute of Technology in 2012. His research interests include visualization and analysis of large-scale scientific data, uncertainty visualization, and flow visualization.



Joachim Moortgat

is an Associate Professor in the School of Earth Sciences, Ohio State University. Moortgat holds MS degrees in theoretical physics and astrophysics, both from Utrecht University, and a PhD in astrophysics from the Radboud University, The Netherlands. Moortgat was the recipient of the 2014 SPE Cedric K. Ferguson Medal awarded by the Society of Petroleum Engineers. His research interests lie in the theory and advanced numerical modeling of compositional multiphase flow in subsurface fractured porous media.



Han-Wei Shen

is a full professor at the Ohio State University. He received his B.S. degree from Department of Computer Science and Information Engineering at National Taiwan University in 1988, the M.S. degree in computer science from the State University of New York at Stony Brook in 1992, and the Ph.D. degree in computer science from the University of Utah in 1998. From 1996 to 1999, he was a research scientist at NASA Ames Research Center in Mountain View California. His primary research

interests are scientific visualization and computer graphics. He is a winner of the National Science Foundations CAREER award and U.S. Department of Energy's Early Career Principal Investigator Award. He also won the Outstanding Teaching award twice in the Department of Computer Science and Engineering at the Ohio State University.