In Situ Feature Analysis for Large-Scale Multiphase Flow Simulations

Soumya Dutta[a], Terece Turton[a], David Rogers[a], Jordan M. Musser[b], James Ahrens[a], Ann S. Almgren[c]

[a]*Los Alamos National Laboratory, Los Alamos, New Mexico, USA*
[b]*National Energy Technology Laboratory, Morgantown, WV, USA*
[c]*Lawrence Berkeley National Laboratory, Berkeley, CA, USA*

Abstract

The study of multiphase flow is essential for designing chemical reactors such as fluidized bed reactors (FBR), as a detailed understanding of hydrodynamics is critical for optimizing reactor performance and stability. An FBR allows scientists to conduct different types of chemical reactions involving multiphase materials, especially interaction between gas and solids. During such complex chemical processes, the formation of void regions in the reactor generally termed as *bubbles*, is an important phenomenon. The study of these bubbles has a deep implication in predicting the reactor's overall efficiency. But physical experiments needed to understand bubble dynamics are costly and non-trivial due to the technical difficulties involved and harsh working conditions of the reactors. Therefore, to study such chemical processes and bubble dynamics, a state-of-the-art computational simulation MFIX-Exa is being developed. Despite the proven accuracy of MFIX-Exa in modeling bubbling phenomena, the large-scale output data prohibits the use of traditional post hoc analysis capabilities in both storage and I/O time. To address these issues and allow the application scientists to explore the bubble dynamics in an efficient and timely manner, we have developed an end-to-end analytics pipeline that enables in situ detection of bubbles, followed by a flexible post hoc visual exploration methodology of bubble dynamics. The proposed method enables interactive analysis of bubbles, along with quantification of several bubble characteristics, enabling experts to understand the bubble interactions in detail. Positive feedback from the experts has indicated the efficacy of the proposed approach for exploring bubble dynamics in very-large-scale multiphase flow simulations.

Keywords: In situ data processing, big data analytics, statistical feature extraction, data reduction, multiphase flow simulation, particle data, feature tracking, HPC, interactive visualization, collaborative development

1. Introduction

With recent advancements in parallel computing capabilities, application scientists are currently building high-resolution computational models to study the working principles of chemical looping reactors (CLR) by simulating various types of multiphase flows. The study of the temporal evolution and dynamics of bubbles in a fluidized bed is of prime interest. In the fluidization process, bubbles (void regions in the fluidized beds) are formed under certain physical conditions, interacting with each other as shown in Figure 1. Understanding the dynamics of such bubbles for these systems is paramount as the formation of large, fast-moving bubbles in fluidized beds causes poor gas/solids mixing, lowering conversion efficiency and stability.

A massively parallel computational fluid dynamics–discrete element model (CFD-DEM) code, MFIX-Exa [1] is being developed by the National Energy Technology Laboratory (NETL) to study multiphase flows in detail. Data generated from MFIX-Exa enables in-depth study of bubble dynamics. However, both computational cost and data size from a single run of the simulation can be large. As a result, the traditional approach of using post hoc analysis and visualization is becoming prohibitively time-consuming and a limiting factor in the ability to derive insight from the data. The bottleneck stems from I/O speed limitations and extreme output data sizes compared to the ever-increasing computing speed. Storing detailed data for individual time steps (e.g., particle positions and velocities in addition to fluid field quantities) is becoming less viable. Scalable and timely visualization and analysis of such data sets pose significant challenges.

The main goal of this work is to develop a practical and reliable solution for domain experts such that they can analyze and visualize bubble dynamics in fluidized

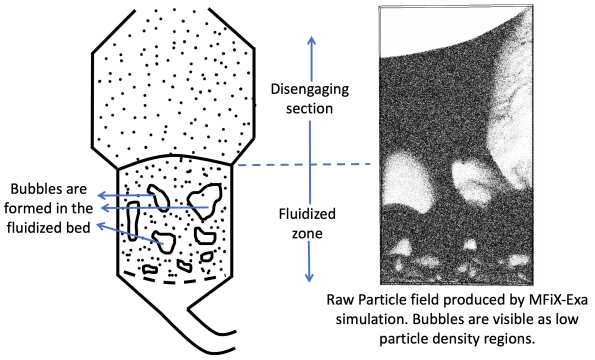

Figure 1: A schematic diagram of a fluidized bed where the bubble phenomenon is shown (left). A visualization of a particle field with bubbles produced by MFIX-Exa simulation (right).

beds in a timely manner using large three-dimensional data sets generated from MFIX-Exa. The experts are interested in understanding the evolution of bubbles and how various characteristics of bubbles evolve. They wish to explore relationships between bubble characteristics and study their velocity profile as bubbles interact with each other. Producing these bubble data sets requires lengthy simulations in supercomputers. The extreme size of the output data is prohibitive to transferring the full data to persistent storage. Therefore, the experts are looking for new solutions that will enable the extraction of important information from the data accurately, leading to a significant amount of storage reduction. Finally, domain scientists need effective and interactive visualization tools that can present the bubble dynamics results via intuitive and simple visual encoding that can be explored, compared, and contrasted interactively and interpreted readily.

In situ analysis techniques provide an attractive solution to address these issues. Since in situ analysis is done as the data is produced in the supercomputer memory, it provides the opportunity to analyze the data in real-time and extract the important information needed to study bubble dynamics. In situ processing can achieve significant data triage and reduction as has been shown recently by many researchers [2, 3, 4]. However, to understand bubble dynamics in detail, expert-in-the-loop interactive visual exploration is necessary where experts can query and filter different bubbles and track them to glean insights. This process is time-intensive and so how much of the proposed analysis pipeline should be done in situ and which analyses should be deferred to the post hoc phase needs to be decided carefully to balance the workload between in situ processing and post hoc analysis.

In this work, we present an analysis technique that combines both in situ and post hoc analysis paradigms resulting in an effective workflow to study bubble dynamics in fluidized beds using data generated from MFIX-Exa simulation. Since the features of interest, the bubbles, referred to as *void regions*, have low particle density, they lack a precise descriptor. In the absence of a precise definition, we employ a statistical distribution-based feature detection technique that has been shown effective for the detection of uncertain features [5, 6, 7]. To reliably detect bubbles, we first compute the particle density field from unstructured raw particles and then model the density field as a homogeneously partitioned distribution field. Finally, the statistical similarity of each partition to that of the target bubble feature is quantified. This similarity-based classification of the particle density field results in a new scalar field called the bubble similarity field (BSF), where each point indicates the possibility of being part of a bubble. Besides the BSFs, to capture the velocity dynamics profile of the bubbles, we also compute another scalar field from the particle velocity data where each point reflects the particle rise velocity. These two derived scalar fields are stored to disk for post hoc analysis.

During the post hoc analysis, the BSFs and the particle rise velocity-based scalar fields (PVFs) are explored in detail to analyze and visualize bubble dynamics. Based on the degree of the similarity values, the BSFs are segmented, and connected component analysis is applied to isolate individual bubbles. For each bubble, several salient characteristics such as bubble aspect ratio, rise velocity, volume, and position are computed. To study the temporal evolution of bubbles, an overlap-based tracking algorithm [8, 9] is applied. An overlap-based tracking algorithm is suitable for our application since we process the data in situ with sufficient temporal frequency to ensure feature overlap between consecutive time steps. Besides tracking individual bubbles, all the bubbles from a specific time step can also be tracked collectively to provide a comprehensive view of the bubble dynamics. The results of computed bubble dynamics and tracking are presented to the experts through Cinema-based interactive viewers [10], which are found to be effective and intuitive in our application study. In addition to the Cinema-based interactive tools, the bubble tracking results are also visualized using an interactive 3D visualization interface so that the users can study bubble evolution directly in the 3D domain. Positive feedback from the domain scientists demonstrates the efficacy of our proposed technique in analyzing bubble dynamics in multiphase flow simulations. Therefore, our contributions in this work are threefold:

C1. We develop a novel in situ analysis workflow for MFIX-Exa simulation and employ a statistical feature detection algorithm to characterize bubble features in the data.

C2. We propose post hoc analysis and visualization techniques that utilize only the reduced in situ generated feature-aware data summaries to extract, isolate, track, and compare bubbles in an effective and timely manner.

C3. We propose a pragmatic and flexible end-to-end feature-driven analytics workflow for domain experts that enables in-depth exploration of bubble dynamics in multiphase flow simulations.

2. Related Work

2.1. In situ analysis

The need for in situ data analysis has grown significantly in recent years to address the problems arising from slow disk I/O. The visualization community has developed several tools for direct in situ rendering of data [11, 12, 13, 14]. All these tools can produce high-quality visualization results in situ. However, when an exploratory analysis is needed, conducting the complete analysis in situ would slow down the simulation significantly. Therefore, a hybrid data analysis paradigm is becoming popular where the data is processed in situ to summarize and extract important information in a compact format and store it to disk for flexible post hoc analysis [3, 15]. This idea has been pursued by many researchers to develop various in situ data summarization algorithms. An image-based in situ data reduction strategy has been shown effective [2, 16]. Statistical distribution-based in situ data summarization techniques have become popular in recent years due to their compactness and flexibility [4, 17, 18]. Statistical downsampling for in situ data reduction has also been explored [19, 20, 21, 22]. An in situ trigger infrastructure has been developed by Larsen et al. [23]. In this current work, we employ an in situ distribution-based feature detection strategy for efficiently classifying bubble features from particle data sets.

2.2. Statistical feature exploration

As the complexity of scientific features has grown, the use of statistical techniques for exploring features in scientific data sets has gained popularity among visualization researchers. Integral histograms were used by scientists to explore local features in scientific data sets [24, 25]. To advance the query-driven visualization capabilities, Gosink et al. [26] used distribution functions for feature analysis. For an enhanced understanding of features in data sets, Johnson and Huang [6] allowed querying on distributions via fuzzy feature matching. Features in ensemble data sets were explored using generalized boxplot-based visualizations [27, 28]. Thompson et al. [29] introduced the idea of hixel, which enabled data summarization as well as the preservation of features in the reduced data sets. Local distribution-based feature extraction and searching was also explored by several researchers [7, 17, 30, 31, 32]. In our work, we follow a similar approach where the data is modeled using local region-wise distribution models, and then bubble features are detected based on statistical similarity.

2.3. Feature tracking

Feature tracking is considered one of the fundamental visualization tasks for analyzing time-varying data features. A significant amount of research has been done on developing different feature tracking algorithms. Silver et al. [8, 33] proposed one of the earliest feature tracking algorithms that used overlap for feature correspondence. An attribute-based feature tracking was proposed by Samatanay et al. [34]. Earth mover's distance was used for feature tracking by Ji and Shen [35]. For tracking features collectively as a group, Ozer et al. proposed techniques for tracking group dynamics [36]. Muelder and Ma introduced a predictor-corrector-based approach for accurate feature tracking. By utilizing global knowledge from all time steps, a merge tree guided feature tracking was proposed by Saikia and Weinkauf [37]. Feature tracking in joint particle/volume data sets was recently proposed by Sauer et al. [38]. Dutta and Shen developed a distribution-based fuzzy feature tracking algorithm for uncertain features [5]. A comprehensive survey of feature tracking can be found in [39]. In this work, we have used an overlapping-based feature tracking algorithm. By sampling the data frequently during in situ processing, we ensure that overlapping-based tracking can be applied in our application to solve the feature correspondence.

3. Application Background and Motivation

3.1. Application Background

Understanding bubble dynamics in fluidized beds is important to scientists studying multiphase flows to design efficient, cost-effective chemical looping reactors (CLR). In a typical CLR, oxygen from a solid

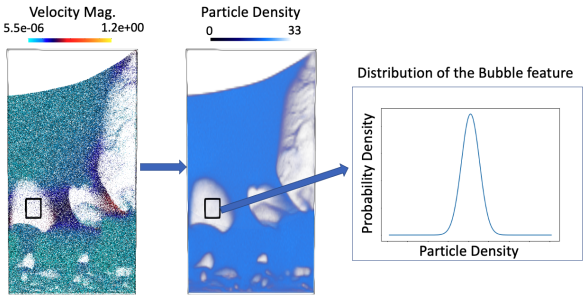

Figure 2: Selection of target bubble feature directly from the data.

oxygen carrier, such as metal oxide, is used to combust fossil fuels [40]. In a standard configuration, the solid oxygen carrier circulates between two fluidized beds, preventing the fuel from directly contacting the air [41]. Large numbers of bubbles can form, causing poor gas/solids mixing and lowering conversion efficiency. Using MFIX-Exa, scientists can analyze and visualize various bubbling phenomena occurring in fluidized beds under different physical conditions. MFIX-Exa is one of the simulation codes in the U.S. Department of Energy's Exascale Computing Project (ECP) [40]. It is expected to achieve exascale performance soon, enabling high-fidelity simulations. Since storing all the high-resolution raw data is not practical due to I/O limitations, developing an in situ analysis capability is crucial for domain scientists. MFIX-Exa simulation code is being built using kernels from the existing MFiX project [42], with a software structure redesign using AMReX [43, 44] as its foundation. AMReX is a block-structured AMR-based software framework designed for building massively parallel applications. MFIX-Exa can produce both mesh and particle data based on the requirements of the application. This work utilized the unstructured raw particle data to study bubble dynamics.

3.2. Motivation

Our motivation comes from surveying existing capabilities and from discussions with the developers of MFIX-Exa about the limitations of current tools for analyzing bubble dynamics. Presently, the MFIX-Exa experts rely on a post-processing workflow with full resolution raw data using tools such as ParaView [45] or VisIt [12]. At current simulation scales, the experts skip on the order of hundreds of time steps when storing data to disk to keep the total data size tractable, possibly missing meaningful events. Given the complex bubble interactions, it is critical to have access to a sufficiently

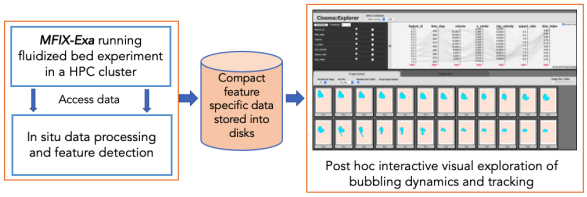

Figure 3: An overview of the proposed end-to-end analysis pipeline.

high temporal resolution of the data so that reliable feature tracking can be done. As the scale of the simulation is expected to grow significantly, the frequency of raw data storage will likely go down even more to address I/O limitations. Post-processing workflows will not scale, making in situ processing critical. Lastly, it can be overwhelming for the domain scientists to analyze and visualize such a large number of bubbles using generic visualization tools as they do not address the specific needs the experts have [46].

4. Domain Requirements and Overview

4.1. Domain Specific Requirements

After discussing with the developers of MFIX-Exa simulation, we have identified several important domain-specific requirements and are listed below:

R1. The storage of full-resolution raw particle data will not be possible with sufficient temporal fidelity, requiring an in situ analysis driven workflow to extract and summarize the important bubble specific information in situ and store reduced data summaries to disk for flexible post hoc analysis.

R2. The experts need the ability to analyze, visualize, and filter bubbles based on various bubble characteristics to explore relationships among different bubble characteristics such as bubble volume, shape, velocity, etc.

R3. The experts want to track the evolution of bubbles from a selected time step to compare and contrast their interactions and dynamics. They want to visualize the velocity profile of the particles around detected bubbles to answer questions such as: Is the rise velocity of bubbles consistent or varying? What is the relationship between bubble volume and rise velocity? How do bubbles merge (or split)?

R4. Domain experts also need customized visualization tools to interactively and intuitively visualize

the analysis results and study bubble dynamics in the 3D spatial domain.

4.2. Feature Selection and Overview of Our Workflow

Since the features of interest lack a precise descriptor, we start by having the users select a region of interest directly in the data offline, based on domain expertise, to highlight a region with low particle density. An example selection is shown in Figure 2. The bubble feature is highlighted using a 3D interactive box filter. Next, a particle density field is estimated from the unstructured particle data, shown in the center image of Figure 2. From this structured density scalar field, all the grid points that fall within the box filter are collected and a Gaussian distribution is estimated using their particle density as shown in the rightmost image of Figure 2. The use of a single Gaussian distribution to model such bubbles is sufficient as the bubbles in the density field generally form a homogeneous region with minimal density variations. This keeps the feature description simple and compact. A mixture of Gaussians can be used if necessary to capture a complex feature as a distribution. This Gaussian distribution is used as a statistical representation of the target feature and statistically similar regions to this distribution can be considered as part of potential bubbles. The use of distributions as a feature descriptor has been shown effective by researchers in the past [5, 7, 48, 49, 50]. Distribution-based feature descriptors are not sensitive to object shape, an advantage in dealing with non-rigid, shape-changing objects such as bubbles [49].

With this distribution-based bubble feature template, detection of statistically similar regions in the data is performed in situ as dicussed in Section 5. The proposed end-to-end analysis pipeline is shown in Figure 3. As can be seen, during the in situ processing, the raw particle data is accessed and the statistical feature detection algorithm is applied to extract bubble features from the particle field compactly and the compact reduced fature-specific data is stored into disks for post hoc analysis. The in situ algorithm generates two scalar fields (BSF and PVF discussed later in Sections 5.3 and 5.4). In the post hoc analysis, the BSFs are segmented, the bubbles are isolated and their various characteristics are measured (Section 6.1). A feature tracking algorithm is employed to track the evolution of bubbles (Section 6.2). All these analysis results are presented to the experts using interactive visualization tools to enable the experts to interact, filter, and track bubbles, exploring how they evolve (Section 6.1). A feature tracking algorithm is employed to track the evolution of bubbles (Section 8). By visualizing the particle rise velocity-based scalar fields (PVFs), they can understand how the particles around the bubbles behave as the bubbles evolve. We also provide summary statistics and scatterplot matrices (SPLOM) to show the relationships between bubble attributes and their time evolution.

5. In Situ Modeling and Bubble Detection

In this section, we describe the in situ data algorithm employed to classify bubbles from the raw particle data, generate a bubble similarity field (BSF), and compute a particle rise velocity field (PVF) used during post hoc analysis. By doing so, we achieve significant data triage and enable flexible and scalable post hoc bubble dynamics analysis.

5.1. Particle Field to Density Field Conversion

In MFIX-Exa, the fluid is model by the particle-unresolved multiphase Navier-Stokes equations which are solved on a structured grid with an embedded boundary formulation used to represent the non-rectangular boundary geometry. Conversely, each particle in the system is modeled as an individual sphere and a linear-spring dashpot (LSD) model is used to compute forces arising from particle-particle and particle-wall collisions. This one-to-one approach to modeling solid particles generates a significant amount of data.

Given this large-scale particle field, since the target bubble features are continuous regions with low particle densities, we first convert the particle field to a density field. Density estimation is often regarded as a fundamental step necessary for sampling particle fields into a structured continuous representation [51]. We have used a spatial histogram-based technique to group particles into non-overlapping bins and then a density field is finally constructed. As the particles are distributed across various computing nodes we compute the histogram in the same distributed setting. A local 3D histogram using particle locations is first constructed at each MPI process by binning the 3D locations of all particles available to each processor. A 3D histogram is required since we are binning particles' 3D locations to estimate spatial particle density. The number of bins and bin widths on each local processing unit is the same and is estimated from the global bounds. We are using uniform bin widths in this work. These global bounds are the bounds of the physical simulation domain, i.e., the domain of the fluidization bed, in which we are creating the 3D histogram. Finally, the partial histograms are combined to construct the global density histogram

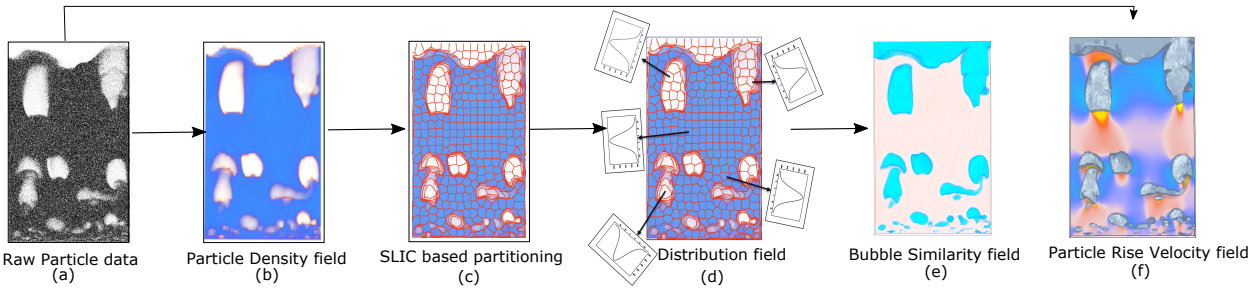

Figure 4: Steps of in situ processing for data transformation, modeling, and bubble detection. We start with the particle data (a) and convert it to a particle density field (b). The particle density field is then partitioned into local homogeneous regions (c), and a distribution field is constructed (d). Finally, a bubble similarity field (BSF) (e) is generated by comparing the distribution field with the user-provided target feature. Besides the BSF, a particle rise velocity field (PVF) (f) is also estimated to analyze the particle velocity profiles during post hoc analysis.

by using a parallel reduction operation overall processing units. Each bin in this global spatial histogram represents particle counts in a local spatial region. The global 3D histogram is transformed into a 3D regular grid-based scalar field where each 3D bin center is mapped to a voxel in the regular grid data and the particle count for that bin is assigned as the particle density value at that voxel. Details of the generation of this regular scalar field from the histogram can be found in [52]. Other particle density estimation techniques [51] can be used in this step, however, we found that using a spatial histogram-based approach to convert the particle data into a density field can be efficiently estimated in situ, keeping the computational cost low during in situ processing. Figure 4b shows an example estimated density field for the raw particle field depicted in Figure 4a.

5.2. Homogeneity-guided Density Field Modeling

Given the lack of a precise descriptor, it is non-trivial to find a consistent hard density threshold value that can be used for isolating low-density regions from the density field. The statistical approach allows the classification of the in situ generated density field into a feature similarity field where regions having high feature similarity values can be explored as bubbles. Since the bubbles are contiguous regions with low particle density, to detect them accurately, we follow a local region-wise statistical data modeling approach to first convert the density field into a distribution field and then classify the distribution field to produce a bubble similarity field (BSF), based on the user-specified target bubble distribution (see Section 4.2).

Our work advocates a clustering-based data partitioning scheme to maximize homogeneous data values in each partition. The high degree of homogeneity allows for more compact distribution-based data modeling with reduced estimation errors. We have used the Simple Linear Iterative Clustering (SLIC) algorithm [53] for producing a homogeneous partitioning of the density field. SLIC is a supervoxel generation algorithm that has been shown to achieve state-of-the-art results both in terms of quality and computational cost.

SLIC is a variant of the local K-means clustering algorithm which works within a predefined local neighborhood while clustering the data. Hence, the total number of distance computations required by SLIC is greatly reduced, resulting in a significant computational speed-up. The algorithm expects the user to specify the approximate size of clusters, $a \times b \times c$. The total number of clusters is estimated and cluster centers are initialized regularly. Since the expected size of a cluster is $a \times b \times c$, the search for similar data points is restricted within a neighborhood $2a \times 2b \times 2c$ around each cluster center [53]. This key strategy significantly reduces the total number of distance computations required. The technique iteratively assigns all the data points to the best representative and when the assignments do not change over consecutive iterations, SLIC terminates. In order to keep a balanced contribution from the data similarity and spatial proximity while assigning data points to a representative cluster, SLIC uses a distance function that considers both the data value similarity as well as their spatial proximity. In this work, we used the distance function as suggested in [54]:

$$D(x, y) = \gamma \cdot \|c_x - p_y\|_2 + (1 - \gamma) \cdot |v_x - v_y| \qquad (1)$$

Here, c_x is the location of the cluster center x and p_y is the location of data point y. v_x and v_y are the scalar values at xth cluster center and yth data point respectively. The value of γ ($0 <= \gamma <= 1$, and $\gamma + (1 - \gamma) = 1$) is chosen based on the requirement to specify weightage for spatial vs value components. We have set $\gamma = 0.2$ for this work to assign higher weightage on data values. Such a distance function ensures that the produced su-

pervoxels are spatially contiguous and as homogeneous as possible. Note that since the global density field is constructed in the root processing node by combining contributions from all other processing units, we compute SLIC partitioning only in the root node using the global density field.

Figure 4c shows the result of SLIC algorithm applied to the density field in Figure 4b. In this example, SLIC is performed in image space for illustration. In in situ, SLIC is performed on the 3D density field. For a more detailed description of SLIC, readers are referred to [53]. After SLIC, density values of data points in each SLIC partition are modeled using a statistical distribution to construct the distribution field, resulting in a compact representation. Since SLIC produces a sufficiently homogeneous partitioning, a single Gaussian distribution for each partition is used to model the density values as shown in Figure 4d. Density values of data points in each SLIC cluster are modeled using a Gaussian and by doing so, the density field is converted to a distribution field.

5.3. Bubble Similarity Field (BSF) Creation

The SLIC-based distribution field is further classified to produce bubble similarity fields (BSF) where high similarity values indicate a higher chance of being part of a bubble. We estimate the density field and collect density values from the user highlighted region (Section 4.2), shown in Figure 2, to define the Gaussian distribution used as the statistical signature of a bubble in the analysis pipeline. During in situ processing, we compute the statistical similarity of each cluster's Gaussian distribution to that of the user-provided signature Gaussian. By doing so, each cluster gets a statistical similarity score. Finally, a new scalar field, called Bubble Similarity Field (BSF), is constructed where each point indicates the degree of statistical similarity to that of the signature bubble distribution. The Bhattacharyya distance [55] is used to measure the similarity between two Gaussian distributions. Since the Bhattacharyya distance provides a closed-form solution and can be computed efficiently, it is found to be suitable for estimating the distance between Gaussians in the in situ environment. The Bhattacharyya distance, $D_{Bh}(g_1, g_2)$, between two Gaussians is defined as:

$$D_{Bh}(g_1, g_2) = \frac{1}{8}(\mu_1 - \mu_2)^T \left(\frac{\sigma_1 + \sigma_2}{2}\right)^{-1} (\mu_1 - \mu_2) + \frac{1}{2} ln\left[\frac{|\frac{\sigma_1+\sigma_2}{2}|}{\sqrt{|\sigma_1||\sigma_2|}}\right] \quad (2)$$

where μ_1, μ_2 and σ_1, σ_2 are the mean and standard deviation of the Gaussian kernels g_1, g_2 respectively. A lower value of $D_{Bh}(g_1, g_2)$ indicates a higher degree of similarity. Before storing values in the similarity field, we normalize the $D_{Bh}(g_1, g_2)$ values and subtract it from 1 to be able to interpret them as similarity scores in post hoc analysis phase. Hence, the values of the BSF range from 0-1 where high values indicate higher statistical similarity to the target feature. Again note that the BSF is computed in the root processing node only and this step does not need any additional data communication. Visualization of a BSF is provided in Figure 4e. It can be seen that the low particle density regions in Figure 4a are classified as the bubbles with high similarity values (the blue regions).

5.4. Particle Rise Velocity Field (PVF) Generation

While studying bubble dynamics, domain experts are also interested in the particle velocity profile around each bubble as they rise through the fluidized bed. Therefore we also estimate a second scalar field which summarizes the vertical rise velocity (x-direction/upward direction) component of the particle velocity field. Since the primary direction of the bubbles are upward, considering just the x-velocity component is sufficient for our analysis. This second scalar field is stored along with the BSF to enable flexible bubble dynamics analysis. The estimation of particle rise velocity scalar field (PVF) follows similar steps that were used for creating the 3D spatial histogram and is again distributed. In this case, the average rise velocity (x-direction) of the particles for each bin is estimated. First, the cumulative rise velocity of all the particles is computed for each bin locally, and then, using a parallel reduction operation, the global velocity histogram is computed. Finally, the average rise velocity per bin is computed by dividing the global 3D velocity histogram by the global density histogram, which is already estimated for computing the particle density field. The particle velocity histogram is then converted to a scalar field following similar steps used for the generation of the density field from the density histogram. A visualization of PVF for the raw particle data, shown in Figure 4a, is provided in Figure 4f. The red and yellowish regions in this image indicate particles with positive rise velocities, i.e., the particles in those regions are rising upward, and the blue regions contain particles that are moving downward. At the end of in situ processing, these two scalar fields are stored on the disk.

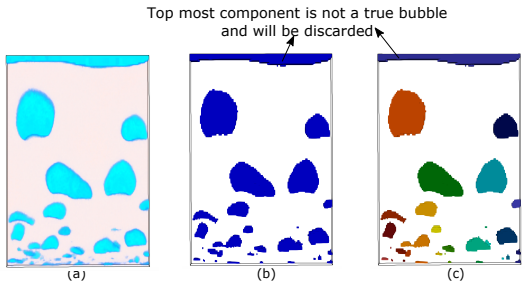

Figure 5: (a) Visualization of bubble similarity field, (b) Segmented bubbles, (c) Individual bubbles as connected components.

6. Post Hoc Bubble Dynamics Analysis

6.1. Bubble Extraction and Characterization

To study details about the bubbles, the first step is to extract individual bubbles from the BSFs. Similarity values in BSF range from 0 to 1 and higher valued regions indicate a higher possibility of being part of a bubble. Experts can inspect several BSFs interactively at the beginning of analysis and specify the desired degree of similarity value to segment the BSFs. This similarity threshold value is applied to all BSFs from all the time steps to produce consistent segmentation results. Next, a connected component algorithm is applied to the segmented results and each component is treated as a separate bubble. Figure 5 shows results of this bubble extraction process for a time step. Figure 5a shows the BSF with blue regions indicating different bubbles. Figure 5b depicts the results when segmentation is performed for a similarity threshold value of 0.92. Finally, results of the connected component algorithm are shown in Figure 5c where different bubbles are colored by their unique component ids. Note that the simulation data often produces a void region on the top of the fluidized bed and has very low particle density. Although it is also detected as one of the potential bubbles (marked in Figure 5b and Figure 5c), these void regions are not considered true bubbles and we ignore this during analysis. Once all bubbles are extracted, several characteristics for each bubble feature are computed. The salient characteristics [46, 47] are considered volume, centroid, and aspect ratio of each bubble. The approach in [34] was used for estimating bubble volume and centroid. For aspect ratio, we used the measure provided in [46].

6.2. Bubble Tracking

A critical requirement of domain experts is to be able to analyze the temporal dynamics of the bubbles on demand. To achieve this, we modified a well-known overlap-based feature tracking algorithm [8] by incorporating an estimation of feature matching confidence to it. In the traditional volume tracking algorithm, the feature correspondence problem was solved by finding overlapping objects in consecutive time steps. This technique assumes the availability of sufficient temporal resolution of data so that overlapping-based correspondence can correctly identify the features over time. However, there exists a minimal chance of incorrect correspondence. To address this issue, we have used the Dice similarity index to detect bubble overlapping in consecutive time steps. Dice index allows estimation of the amount of overlap between two sets with a minimum of 0 indicating no overlap, and a maximum value of 1 for complete overlap. Formally for two sets, A and B, their Dice index $DI(A, B)$ is measured as:

$$DI(A, B) = \frac{2\,|A \cap B|}{|A| + |B|} \tag{3}$$

The use of the Dice index allows us to detect the overlap between two segmented bubbles in 3D and also quantify a similarity score reflecting the degree of matching. The Dice index value can be interpreted as matching confidence and when an abruptly low value is found, such a time step is flagged for further investigation. In our visualization tool, values of the Dice indices are presented so that the experts can make informed judgments on the tracking results. In bubble evolution, it is important to study the different evolutionary events that the bubbles go through such as splitting and merging with neighboring bubbles. In a typical fluidized bed simulation, small bubbles form at the bottom of the bed, merging and splitting as they rise through the fluidization zone before bursting out at the top. In our method, we track bubble volumes to explore potential merge and split events. During tracking, a sudden rise in the bubble volume indicates a merge event while a significant drop in volume would indicate splitting [5]. From the tracking results, we also estimate the bubble rise velocity, an important bubble characteristic of interest to the domain experts. The bubble rise velocity is computed from differences of the centroids of matched bubbles from the consecutive time steps.

7. Visual Exploration of Bubble Dynamics

Next, we present bubble dynamics through several interactive visualization interfaces such that the experts can query, filter, and track bubbles to understand their interaction. In this work, we have used CinemaExplorer [10] tool to visualize image-based bubble dynamics. Note that we use this image-based approach for

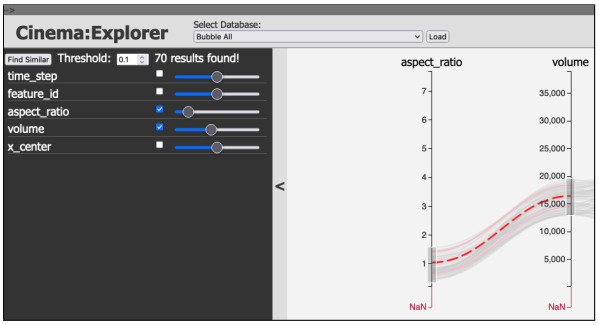

Figure 6: Interactive query of multiple bubble characteristics using the query panel of CinemaExplorer. Bubbles that fall within a range of aspect ratio and volume are queried in this example.

the experts to quickly and interactively study the bubble dynamics. However, we allow detailed investigation of bubbles and their tracking in the 3D domain when desired.

In our simulation test cases, the Y-dimension is typically smaller compared to the other two dimensions (X and Z), so the experts visualize the bubbles from the X-Z plane, which maximizes the viewing area of the simulation domain. Therefore, after bubbles are extracted, we use volume rendering algorithm to render each bubble viewed from the X-Z plane and store visualizations as images into a database following the Cinema specifications [56]. First, we visualize the field in ParaView [45] and generate a Python-based ParaView rendering script containing the rendering specifications and then run the script to produce images of all the bubbles in a batch mode. All rendering parameters (viewing and camera parameters, and transfer functions for volume rendering) are kept fixed. Note that this operation is done once and takes only ~5 minutes to produce all the images. This results in an image-based database that includes bubble characteristics and a visual representation of the bubbles. The intuitive and well-known visualization techniques used in CinemaExplorer, such as Parallel Coordinates Plots (PCP) and Scatterplots are used to present the results to the domain experts. Along with these interactive tools, we have also developed a visualization tool that allows visualization of bubble tracking directly in the three-dimensional domain for the experts to study bubble interaction in 3D.

7.1. Overview Visualization of Bubble Characteristics

The first step in our visual analysis pipeline uses CinemaExplorer to study overall bubble characteristics so that the experts can obtain a general understanding. A typical CinemaExplorer viewer has three panels as highlighted in Figure 7: (A) the query panel, (B) the Parallel Coordinates Plots (PCP) panel, and (C) image spread panel. The query panel (A) provides multivariate query capabilities as shown in Figure 6. In Figure 6, data is filtered using aspect ratio and volume attributes. The red dotted line shows the central value around which the filtering is done, and the dotted red lines show the filtering range. Panel (B) shows the PCP where the bubble characteristics are represented as parallel axes and the characteristics of each bubble are represented by a polyline. The PCP supports interactive brushing and axis reordering. The visualization of the bubbles is shown in the image spread view, Panel (C). The image size can be adjusted using a slider on the top right corner of the panel. Filtering in PCP or query panel automatically updates results in the image view. An additional tab provides a scatterplot visualization of any two selected bubble characteristics (see, e.g., Figure 8). Such scatterplots are found to be effective in displaying the temporal evolution of bubble characteristics. This analysis tool and capabilities address the **R2** of the domain experts mentioned previously in Section 4.1.

7.2. Visualization of Bubble Tracking

7.2.1. Visualization Using Cinema-based Viewer

Once users identify an interesting bubble, it is tracked over time and the analysis result is presented using CinemaExplorer as shown in Figure 7. In this example, a relatively large bubble was selected from time step 116 and tracked both forward and backward in time. Note that the PCP has two axes showing information about the rise velocity and the values of the Dice index estimated during tracking. The Dice index values are mostly high, indicating stable tracking results. Using this visualization, the experts can interactively explore the life cycle of the selected bubble. The CinemaExplorer image spread panel presents the visualization of the bubbles side-by-side. As discussed above, users can also filter results based on the bubble attribute values by using both the PCP and the query panel. The scatterplot functionality in the CinemaExplorer viewer is useful to investigate the details of the various evolutionary events such as merging and splitting. In Figure 8, the scatterplot of bubble volume vs time steps is shown at the top. In the simulation, bubbles are generally formed at the bottom of the bed and rise upward, moving through the fluidized bed. Time steps when bubble volumes abruptly change are easily identified in the scatterplot. Several such events are highlighted in the plot with the specific events that caused the change in bubble volume. The visualization of these events for this bubble is provided at the bottom of Figure 8. It can be seen that,

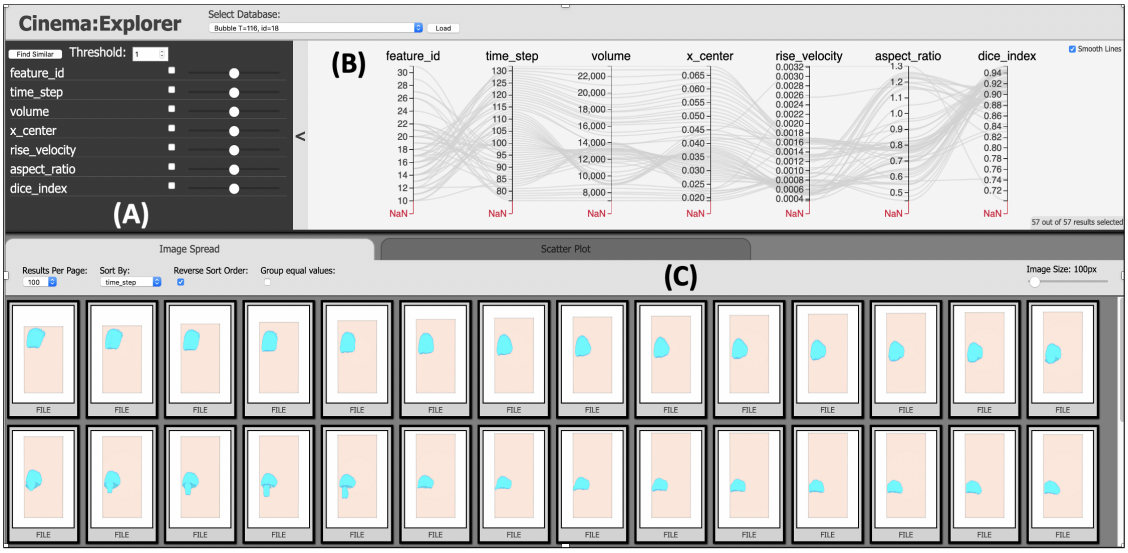

Figure 7: Visualization of bubble characteristics and tracking using CinemaExplorer interface. The PCP shows relevant bubble characteristics such as aspect ratio, position of bubbles in the rising direction, time step, etc. Visualizations of tracking of a specific bubble is shown at the bottom panel. This initial bubble was selected from time step = 116 with feature id = 18 for tracking.

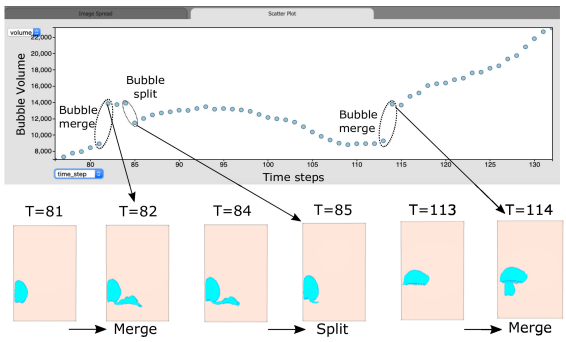

Figure 8: (Top) Volume vs time scatterplot revealing bubble merge and split events with the initial bubble selected at time step=116 and feature id=18. (Bottom) Detected events are shown for different time steps.

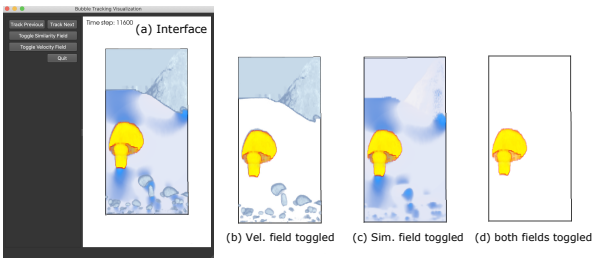

Figure 9: Visualizing bubble tracking directly in 3D domain using our feature tracking interface shown in Figure 9(a). The interface shows the tracked bubble (highlighted in yellow). All the other bubbles are shown as context and, in the background, the regions that have high particle rise velocity is also presented (colored in light blue). The interactive interface allows the users to turn on and off different visualization components. In Figure 9(b) PVF is turned off, in Figure 9(c) the bubbles are turned off, and Figure 9(d) depicts the visualization of only the tracked bubble.

indeed, the abrupt increase in bubble volume indicates merge events and split events correspond to a sudden decrease in bubble volume.

7.2.2. Visualizing Bubble Tracking in 3D Domain

As mentioned above, many characteristics of the tracked bubbles can be analyzed using the CinemaExplorer tool. However, to allow the users to explore the 3D nature and shape of the bubbles, we have developed a feature tracking visualization tool that uses 3D volume rendering techniques to provide interactive visualization of bubbles. This step also satisfies the expert requirement **R4** as mentioned before in Section 4.1. This tracking interface is shown in Figure 9(a). The interface integrates VTK [57] and QT for rendering and interaction capabilities. The users can interactively change time steps. In this interface the tracked bubble is highlighted in yellow. In the background, all the other bubbles are shown with grey color for context. Finally, the regions where the PVF have high values, i.e, the regions where the particles are moving faster than others are shown using volume rendering and are colored in light blue as seen from Figure 9(a). Different visualization components convey different information and the combined view provides a holistic understanding for

the users who can study the temporal bubble evolution while also visualizing the dynamic nature of the particle velocities around the bubbles. At any time, the users can interactively turn on or off the additional bubbles or the PVF visualization to focus on the tracked bubble. In Figure 9(b), we show the rendering window when PVF is turned off, in Figure 9(c) the additional bubbles are turned off, and finally in Figure 9(d) only the tracked bubble is shown.

7.3. Scatterplot Matrices for Correlation Study

Besides visualizing the bubbles through the CinemaExplorer tool and our 3D tracking interface, the users are also interested in discovering correlations among various bubble characteristics, noted as requirement **R3**. To achieve this, we provide a scatterplot matrix, (SPLOM), visualization for the users where all pair scatterplots are shown simultaneously in a matrix-based layout (as shown in Figure 11). SPLOM gives a quick overview visualization where the users can inspect correlations between any two bubble characteristics. Once they find an interesting pair, they can use the scatterplot in CinemaExplorer for a detailed study where the users can interactively highlight points in scatterplot and inspect the corresponding bubble characteristics values. Therefore, by combining CinemaExplorer functionality, our feature tracking interface, and SPLOMs, we are able to present in-depth information about the temporal bubble dynamics to the application experts.

8. Results

We validate the above findings on bubble dynamics through expert feedback and by comparing our results with findings in the multiphase flow research literature. We provide details of bubble evolution from the tracking results and explore the relationships among various bubble characteristics. Note that, we have linearized the time step numbers during analysis. The actual simulation time step numbers can be found by multiplying 100 with the numbers reported here.

8.1. Analysis of Bubble Characteristics

Bubbles in the fluidized bed form consistently at the bottom of the bed and rise upward before finally reaching the bed surface. Initially, bubbles are generally small, and gradually grow as they rise, merging with other bubbles. After studying bubble tracking results for multiple bubbles, we observed that bubble merge events are more frequent than bubble splitting and as a result, the majority of the bubbles grow in size over time before reaching the top. This observation was intriguing to the domain expert and the expert was able to reason about such phenomenon. It is also noted that while studying bubble tracking results using CinemaExplorer, the scatterplot between bubble volume and time steps effectively shows this consistent trend for the majority of the bubbles studied. Figure 10 illustrates how bubble volume changes over time with sudden changes signifying merge/split events. Also, during the investigation of these events, by filtering the bubbles using aspect ratio (width/height) values, users can easily locate bubbles that are spherical in shape (aspect ratio ≈ 1). Such bubbles are often of interest to the experts [46]. We further observe that the bubble rise velocity increases slowly as the bubble volume increases. Consistent with existing literature, bubble rise velocity remains relatively constant if bubble volume remains constant [46], whereas rise velocity increases with increasing volume [46, 58]. Hence, our method helps in the validation process of the simulation code.

SPLOM-based correlation study. In Figure 11, a SPLOM is shown for a specific bubble tracked over time. The SPLOM is plotted as a lower triangular matrix where each cell is a scatterplot between two specific bubble characteristics. The points in the SPLOM scatterplots are colored by the Dice similarity index, indicating the bubble matching confidence during tracking at each time step. For comparative analysis, we also provide the PCP from CinemaExplorer and three representative visualizations of the bubble, one from an initial time step (T=107), the second from an intermediate time step (T=137), and the third from a later time step (T=154). By observing the SPLOM, we find that the bubble volume and rise velocity increases with time (Cell [1,0], and [3,0] in SPLOM). Furthermore, the rise velocity and bubble volume (Cell [3,1]) is also correlated directly. Similar relationships among bubble volume and rise velocity were documented in previous studies [46, 58]. This correlation between bubble volume, rise velocity, and time is also observed in the PCP in Figure 11 among PCP axes time step, volume, and rise velocity respectively. Note that the axis x_center indicates the position of the bubble in the rising direction and since the bubble rises gradually over time, x_center is also found to be correlated with volume and rise velocity as seen from both SPLOM and PCP.

8.2. Particle Velocity Dynamics Exploration

The study of particle dynamics around the bubbles is a challenging and key task for domain scientists as

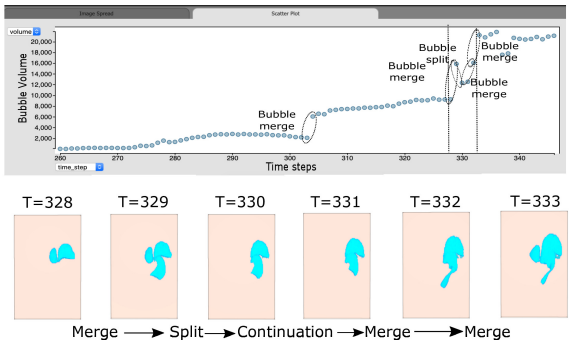

Figure 10: (Top) Volume vs time step Scatterplot revealing bubble merge and split events. The initial feature was selected from time step = 338 with feature id = 7. (Bottom) Visualization of merge and split events for this bubble. Detected merge and split events are shown for a time window indicated by vertical dotted lines in the Scatterplot.

it allows exploration of the hydrodynamics in the fluidized bed [46]. Visualization of particle rise velocity fields (PVF), computed in situ, complement the above analyses, helping to explain particle dynamics around the bubbles. Figure 12 shows a PVF for four different time steps representing different states of the simulation. The selected colormap shades particles moving upward (positive velocity) red-to-yellow whereas downward-moving particles around each bubble are blue. An important hypothesis in this regard is that particles around the bubbles tend to move downward as was reported in [46]. Our domain expert wanted to see if that hypothesis is followed in MFIX-Exa and Figure 12 shows indeed that is the case. The rationale behind such a phenomenon is that this happens so that the particles around the bubble can move down and replenish those carried upward in the wake of the bubbles. The wake of a bubble is the region immediately behind it (the red-to-yellow region in the images). It is also observed that particle velocities above and below bubbles are high. The velocity is higher underneath bubbles in the wake than above bubbles which is seen from the yellow regions at the bottom of bubbles. This distribution of low-velocity particles around the bubbles and high-velocity particles on the top and bottom of the bubbles generates a circular flow causing the bubbles to rise before breaking through the freeboard. Our analysis method shows this phenomenon visually and the domain experts found this extremely useful for validating and verifying the working principles of MFIX-Exa.

9. Discussion

9.1. Expert Feedback

Results were presented to the domain experts who are developers of MFIX-Exa and are co-authors of this paper. Overall, the experts were very excited to see the broad and comprehensive capability that we have developed. Before this work, most of the analyses were done offline using visualization tools such as ParaView and were time-consuming. The experts found the three-dimensional bubble visualization capabilities very useful, allowing them to investigate the bubble evolution directly in 3D. The experts were excited to learn about the novel in situ workflow enabling the detection of bubbles in situ followed by the interactive, flexible, and real-time post hoc analysis of bubble dynamics which satisfied their critical need of a flexible analysis workflow, noted previously as a requirement **R1** in Section 4.1. They agreed that this new analysis capability will accelerate their scientific discovery process while analyzing simulation data and felt that this workflow is a key for them to be able to interactively explore full-scale three-dimensional bubble dynamics. The experts were also comfortable with CinemaExplorer, which consisted of well-known visualization techniques, and found them intuitive and useful. They felt that the 3D bubble tracking interface worked as a complementary tool to CinemaExplorer, allowing them inspection of bubbles in the 3D domain. Through the PCP, they could filter interesting bubbles based on the various salient characteristics. The addition of SPLOM was also found to be effective for validating relationships between various bubble characteristics. Overall, the experts were very positive about our approach and thought that our work has made significant contributions in the analysis of three-dimensional bubble dynamics for multiphase flow simulations.

9.2. Parameter Selection

The cluster size for SLIC is an important parameter. Generally, a smaller cluster size improves accuracy since smaller clusters will detect smaller bubble regions more accurately; if the cluster size is too large, some small bubbles could be missed. Figure 13 shows the impact of two different cluster sizes. It can be seen that with a large cluster size, some small bubbles are fragmented (highlighted in red). Hence, to capture small bubbles accurately, we have used 3x3x3 cluster sizes in all our experiments. Finally, for segmenting bubbles from the BSFs, we have used a consistent statistical similarity threshold of 0.92, which resulted in consistent

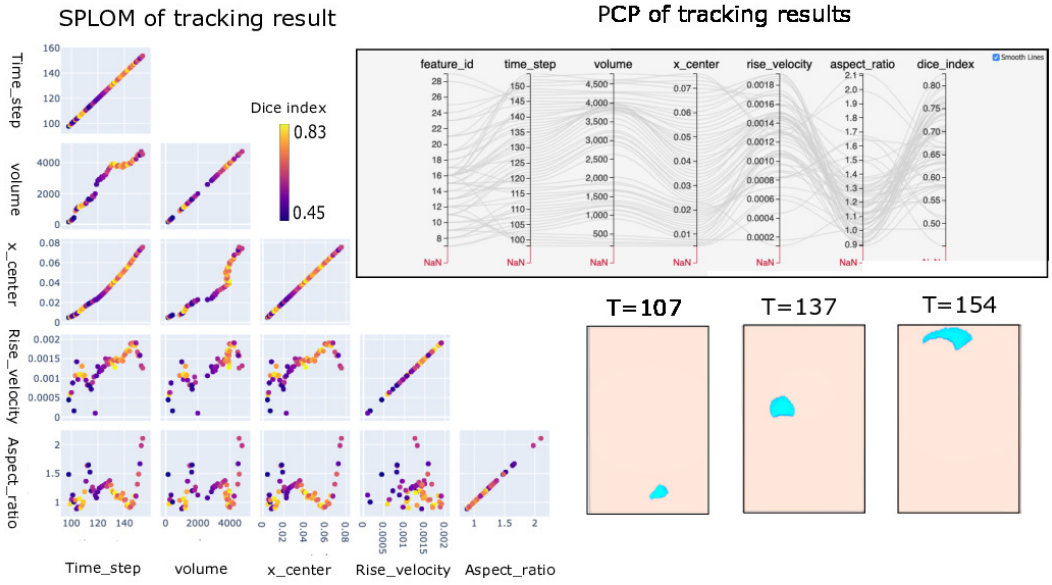

Figure 11: Visualization of SPLOM generated for studying relationships among various salient bubble characteristics for a bubble tracked over time. The bubble was selected from time step = 154 with feature id = 16. The PCP from the CinemaExplorer is also shown. At the bottom right, visualizations of the selected bubble from three different time steps are provided. It can be seen that the bubble increases in size over time as it evolves through the fluidized bed.

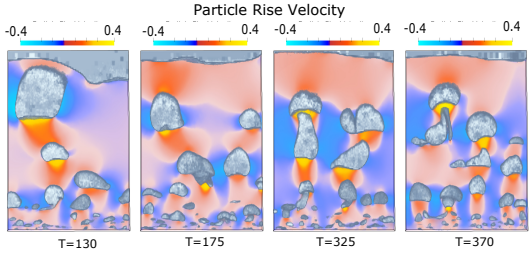

Figure 12: Visualization of particle rise velocity fields (PVF) for different time steps. The blue regions indicate particles with negative velocity (downward movement) and the red to yellow regions show particles with positive rise velocity(upward movement). The PVFs are computed in situ from the raw particle velocity fields and can be used to effectively study the dynamics of particles around bubbles.

bubble extraction for all the time steps for all simulation use cases. By visually comparing the segmented bubbles with the raw particle fields, it was found that the extracted bubbles accurately represented the void regions. We also found that changing the similarity value slightly does not significantly change the shape of the bubbles and so the segmentation is robust.

In this context, one might just use simple thresholding to extract bubbles directly from the particle density field. However, finding a robust and consistent threshold that will work for different simulation cases is non-

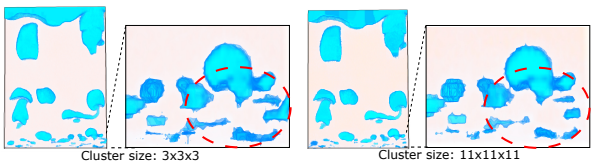

Figure 13: Impact of different cluster sizes on bubble detection. For larger cluster sizes, small bubbles might be detected inaccurately.

trivial since there is no guideline for picking a density threshold. Also, for different simulation use cases, the dynamic range of density fields will be different and so the same threshold will not work across simulations. In Figure 14, we show that when we use a density threshold = 12 for extracting bubbles, while it works reasonably well in one use case (Figure 14e), the same threshold does not work for the other use case (Figure 14b). In contrast, our statistical method uses the same feature similarity threshold for extracting bubbles for both of the use cases and can extract bubbles correctly as shown in (Figure 14c and Figure 14f).

9.3. Comparative Discussion with Similar Works

Bubble feature analysis and tracking in a two-phase simulation data were also explored by Fang et al. [59]. In this work, as the simulation was able to generate

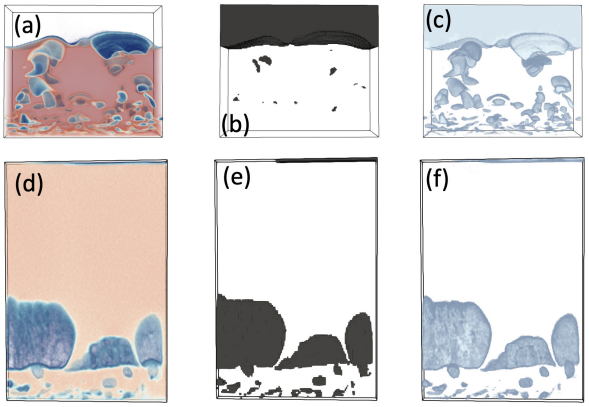

Figure 14: Comparing density threshold-based bubble extraction with the proposed method. The top row ((a), (b), and (c)) and the bottom row ((d), (e), and (f)) show two different simulation use cases. Finding a robust and fixed density threshold to segment bubbles is not possible since different simulations will have varying density value ranges and the same density threshold will not work. We see that the density threshold=12 works well for one case, (e), but the same threshold fails in the second case, (b). In contrast, the proposed method is able to use the same feature similarity to extract bubbles as shown in (c) and (f).

a level crossing scalar field data that indicated separation between two states, the researchers used a level set-based method to extract bubbles from the data and then tracked them over time. The boundary of the bubble was isolated by the level set of zero values in the field. In our case, since we do not have this kind of level crossing field and rely on the unstructured particle data, we are not able to use a level set-based method. Instead, we use a statistical distribution-based method to reliably extract bubbles from the particle field. In another recent post-processing pipeline, an analysis of bubbles was proposed by Buchheit et al. [47]. It used mesh-based data generated from MFiX simulation, the predecessor of MFIX-Exa simulation. Bubbles were identified by a constant threshold in the volume fraction field and their contours (level set) were visualized and analyzed. Although this technique shows promise, using a fixed threshold to segment bubbles could lead to missing smaller bubbles. Furthermore, while tracking, the intersection between corresponding bubbles was tested by intersecting the bounding boxes which also could result in incorrect correspondence for convex bubbles [47]. We remedy these shortcomings and sample the data frequently in situ to ensure that overlap-based reliable bubble tracking can be applied. The overlap test is done on the 3D bubble features and hence the correct overlap is detected. We employ a statistical feature detection technique that generates a bubble similarity field which

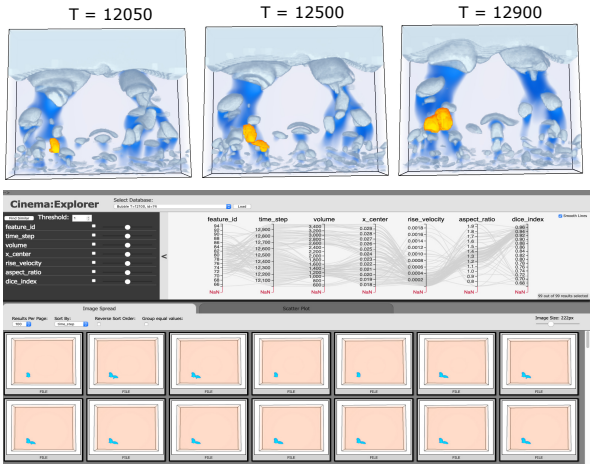

Figure 15: Visualization of bubble tracking results from the large-scale in situ MFIX-Exa use case. The feature similarity fields and particle velocity fields shown in the top row are generated in situ.

can be visualized with varying similarity values to illustrate uncertainties associated with bubble detection. Hence, in our case, we are not using a level set to extract bubbles. We also provide customized visualization tools that effectively show the results of bubble dynamics interactively, enabling the user to query bubble properties and study tracking results.

10. In Situ Case Study

Until now, we have provided a detailed study of our proposed algorithm and showed its efficacy in exploring bubble dynamics. In this section, we discuss the details of an in situ case study and its computational performance to demonstrate the practicality and in situ viability of our technique. The MFIX-Exa simulation data that we have used above for demonstration purposes, contains ≈3 million particles per time step. We have used this data set for developing, benchmarking, and validating our proposed technique by working closely with the MFIX-Exa developers. For the in situ performance study, we have used a larger MFIX-Exa use case. This new use case simulates 54.51 million particles in a fluidization bed. In this controlled study, we simulate 1000 time steps for measuring in situ performance and storage benefits of our proposed technique. In Figure 15, we show results from this large-scale simulation where the top row shows 3D visualization of tracking of a specific bubble from three representative time steps, and at the bottom, we show the CinemaExplorer-based visualization for this bubble. The BSF and the PVFs

shown in the top row are generated in situ and our tracking interface is used to generate these visualizations.

10.1. In Situ Code Integration

The MFIX-Exa [60] and AMReX [44] codes are developed in C++. To perform in situ data analysis, custom code was added to both MFIX-Exa and AMReX code bases. The in situ code is developed in C/C++ and uses VTK data models [57]. The final summarized bubble similarity fields and the particle rise velocity fields are stored in VTK image data format, and so, can be readily loaded into standard visualization tools such as ParaView [45] or VisIt [12]. A direct data access scheme is used to read the raw particle data from memory by querying the particle container data structures of AMReX. Since the simulation and the in situ data processing routines use the same memory and computing resources, the proposed in situ integration works in synchronous mode, tightly coupled with the simulation. As we output the in situ summarized feature-specific data in the form of 3D scalar fields that can be analyzed and visualized to produce new results, the output type of our in situ processing is explorable.

10.2. In Situ Performance Evaluation

We deployed our algorithm in situ by running it with MFIX-Exa simulation using a moderately sized test case. The primary purpose of this performance evaluation is to measure how much extra time our algorithm takes compared to the actual simulation and how much storage reduction our method can achieve. Another purpose is to estimate how much I/O cost and post hoc analysis time we can save by performing part of the analyses in situ. This in situ performance evaluation was done the Summit supercomputer [61], an IBM system located at the Oak Ridge Leadership Computing Facility. Below we show the results we obtained from the performance evaluation in terms of space and time savings.

Storage savings. A single time step of this MFIX-Exa test case outputs 3.1 GB of raw data for three quantities: particle id, location, and velocity. Note that the simulation typically stores many other fluid variables per particle, such as drag, mass, omoi, density, phase, volume, etc., which will increase the storage per time step further. However, since we are only using particle location and velocity variables, we only store these variables so that we get a fair storage comparison. We simulated 1000 time steps, producing over 3 TBs of data. Due to the high data volume, experts often only store every 100th time step so the resultant data size remains tractable. Note that, for the purpose of our study, we only simulated 1000 time steps, however, for the real analysis case, typically tens of thousands of time steps are needed to run and the storage requirement will increases proportionately. Processing data in situ allows us to access higher temporal fidelity so that more time-accurate information about the bubbles can be captured. In our experiment, we accessed simulation data every 10th time step for 1000 total time steps, resulting in storage of 100 time steps. Storing raw data for these 100 time steps requires 310 GB storage space. Instead, by performing in situ analysis, for every 10th time step, we computed bubble similarity and particle rise velocity fields and stored those fields to disk. The disk space required for storing the in situ generated data is only 224 MB in VTK format which is significantly smaller compared to the raw particle data storage. Hence, we find that in situ analysis workflow is beneficial and results in a significant data triage while paving a path for flexible and detailed post hoc study of bubble dynamics.

Computational time savings. Since the simulation starts from a random particle initialization and gradually reaches a state where bubbles are periodically formed, we started MFIX-Exa simulation from a later time point (time step = 12000) to produce bubbles and simulated 1000 time steps for our study. In Table 1, we provide the computational and I/O timings taken by the different steps of the algorithm and the study was run using 2048 MPI processes. The first three columns of Table 1 show computation times of density and PVF field estimation, SLIC generation, and computation of similarity fields separately. We find that the total in situ processing time is significantly smaller compared to the actual simulation time and the in situ processing takes only around 3% additional computation time. Hence, our in situ algorithm does not overburden the simulation. The last two columns show the I/O timings of in situ processing and the time if the raw particle data was stored. We observe that the reduced in situ outputs take significantly less time to be stored into disks compared to the time taken by raw particle field storage. Hence, in situ processing also minimizes the disk I/O time.

In Table 2, we show the time it would take if our algorithm was run offline using stored raw particle data on a standard desktop computer. We see that the off-line processing of such a large time-varying data set takes much longer time compared to the time taken in situ (Table 1). The density and velocity field computation time takes the largest fraction of computation time since this computation needs the raw particle data. The computation of SLIC and similarity field uses the derived particle density field which is smaller than the raw data and hence the computation faster. We can use parallel

Table 1: In situ processing and I/O times (in seconds) taken by the proposed method compared to the simulation time.

	Density and velocity field	SLIC	Similarity field	Total in situ computation	Total simulation time	In situ I/O	Simulation I/O
2048 MPI processes	2.58	124.37	1.44	128.39	4408.6	14.60	504.85

Table 2: Post hoc timings (in seconds) for different steps of our proposed algorithm. By processing data in situ, timings shown in this table can be saved.

Density and velocity field	SLIC	Similarity field	Total I/O
39420.87	59.27	144.57	3364.43

processing to improve upon these computation timings, however, the I/O time (third column of Table 2) will still be high for post hoc analyses, and as the number of time steps to process will increase, these post hoc computation timings will become significantly high. In contrast, by processing data in situ, we can completely bypass timings reported in Table 2 and the domain experts can start their analysis right after the simulation ends and the exploration will be accelerated.

11. Conclusions

In this work, we have presented an end-to-end in situ analysis guided visual exploration of complex bubble dynamics phenomena in fluidized bed simulations. We have successfully demonstrated an in situ analysis pipeline with MFIX-Exa to extract bubble-specific data in situ, enabling flexible and scalable post hoc exploration of bubble dynamics. We have conducted a detailed performance study and validated our findings through domain expert feedback and qualitative comparison with existing literature in the multiphase flow analysis. In the future, we plan to apply our technique to ensemble MFIX-Exa simulations so that the role of input parameters in the bubble dynamics can be explored efficiently. We also want to explore other particle density estimation techniques provided in [51] for computing particle density fields. Finally, GPU implementations of our in situ algorithm are underway to further improve in situ performance of future extreme-scale MFIX-Exa runs with hundreds of millions of particles at upcoming exascale supercomputers [40].

Acknowledgments

The authors would like to thank the Department of Energy and Los Alamos National Laboratory for the funding and support in carrying out this research. This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration and is released under LA-UR-22-24574. This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

References

[1] MFIX-Exa, https://amrex-codes.github.io/MFIX-Exa/docs_html/ (2022 (March 23, 2022)).
URL https://amrex-codes.github.io/MFIX-Exa/docs_html/

[2] J. Ahrens, S. Jourdain, P. OLeary, J. Patchett, D. H. Rogers, M. Petersen, An image-based approach to extreme scale in situ visualization and analysis, in: SC14: International Conference for High Performance Computing, Networking, Storage and Analysis, 2014, pp. 424–434. doi:10.1109/SC.2014.40.

[3] C. Wang, H. Yu, K. L. Ma, Application-driven compression for visualizing large-scale time-varying data, IEEE Computer Graphics and Applications 30 (1) (2010) 59–69. doi:10.1109/MCG.2010.3.

[4] S. Dutta, C. Chen, G. Heinlein, H.-W. Shen, J. Chen, In situ distribution guided analysis and visualization of transonic jet engine simulations, IEEE Transactions on Visualization and Computer Graphics 23 (1) (2017) 811–820.

[5] S. Dutta, H.-W. Shen, Distribution driven extraction and tracking of features for time-varying data analysis, IEEE Trans. on Vis. and Comp. Graphics 22 (1) (2016) 837–846. doi:10.1109/TVCG.2015.2467436.

[6] C. Johnson, J. Huang, Distribution-driven visualization of volume data, IEEE Trans. on Vis. and Comp. Graphics 15 (5) (2009) 734–746. doi:10.1109/TVCG.2009.25.

[7] C. Lundstrom, P. Ljung, A. Ynnerman, Local histograms for design of transfer functions in direct volume rendering, IEEE Trans. on Vis. and Comp. Graphics 12 (6) (2006) 1570–1579. doi:10.1109/TVCG.2006.100.

[8] D. Silver, X. Wang, Volume tracking, in: Proceedings of the Visualization '96 Conference, Computer Society Press, 1996, pp. 157–164.

[9] C. Muelder, K.-L. Ma, Interactive feature extraction and tracking by utilizing region coherency, in: Visualization Symposium, 2009. PacificVis '09. IEEE Pacific, 2009, pp. 17–24. doi:10.1109/PACIFICVIS.2009.4906833.

[10] Cinema:Explorer, a general viewer for Cinema, Spec D, https://github.com/cinemascience/cinema_explorer (2022 (accessed March 23, 2022)).

[11] N. Fabian, K. Moreland, D. Thompson, A. C. Bauer, P. Marion, B. Gevecik, M. Rasquin, K. E. Jansen, The ParaView coprocessing library: A scalable, general purpose in situ visualization library, in: 2011 IEEE Symposium on Large Data Analysis and Visualization (LDAV), 2011, pp. 89–96. doi:10.1109/LDAV.2011.6092322.

[12] B. Whitlock, J. M. Favre, J. S. Meredith, Parallel in situ coupling of simulation with a fully featured visualization system, in: Proceedings of the 11th Eurographics Conference on Parallel Graphics and Visualization, EGPGV '11, Eurographics Association, 2011, pp. 101–109. doi:10.2312/EGPGV/EGPGV11/101-109.

[13] M. Larsen, J. Ahrens, U. Ayachit, E. Brugger, H. Childs, B. Geveci, C. Harrison, The alpine in situ infrastructure: Ascending from the ashes of strawman, in: Proceedings of the In Situ Infrastructures on Enabling Extreme-Scale Analysis and Visualization, ISAV'17, Association for Computing Machinery, New York, NY, USA, 2017, p. 42–46. doi:10.1145/3144769.3144778.

[14] SENSEI:Scalable in situ analysis and visualization, https://sensei-insitu.org (2022 (accessed March 23, 2022)).
URL https://sensei-insitu.org

[15] H. Lehmann, B. Jung, In-situ multi-resolution and temporal data compression for visual exploration of large-scale scientific simulations, in: IEEE 4th Symposium on Large Data Analysis and Visualization (LDAV), 2014, 2014, pp. 51–58. doi:10.1109/LDAV.2014.7013204.

[16] A. Tikhonova, C. Correa, K. L. Ma, Explorable images for visualizing volume data, in: 2010 IEEE Pacific Visualization Symposium (PacificVis), 2010, pp. 177–184. doi:10.1109/PACIFICVIS.2010.5429595.

[17] K. Wang, K. Lu, T. Wei, N. Shareef, H.-W. Shen, Statistical visualization and analysis of large data using a value-based spatial distribution, in: 2017 IEEE Pacific Visualization Symposium (PacificVis), 2017, pp. 161–170.

[18] Y. C. Ye, T. Neuroth, F. Sauer, K.-L. Ma, G. Borghesi, A. Konduri, H. Kolla, J. Chen, In situ generated probability distribution functions for interactive post hoc visualization and analysis, in: 2016 IEEE 6th Symposium on Large Data Analysis and Visualization (LDAV), 2016, pp. 65–74.

[19] T. Wei, S. Dutta, H.-W. Shen, Information guided data sampling and recovery using bitmap indexing, in: 2018 IEEE Pacific Visualization Symposium (PacificVis), 2018, pp. 56–65.

[20] J. Woodring, J. Ahrens, J. Figg, J. Wendelberger, S. Habib, K. Heitmann, In-situ sampling of a large-scale particle simulation for interactive visualization and analysis, in: Proceedings of the 13th Eurographics / IEEE - VGTC Conference on Visualization, Eurographics Association, 2011, pp. 1151–1160. doi:10.1111/j.1467-8659.2011.01964.x.

[21] T. Rapp, C. Peters, C. Dachsbacher, Void-and-cluster sampling of large scattered data and trajectories, IEEE Transactions on Visualization and Computer Graphics 26 (1) (2020) 780–789.

[22] A. Biswas, S. Dutta, E. Lawrence, J. Patchett, J. C. Calhoun, J. Ahrens, Probabilistic data-driven sampling via multi-criteria importance analysis, IEEE Transactions on Visualization and Computer Graphics 27 (12) (2021) 4439–4454. doi:10.1109/TVCG.2020.3006426.

[23] M. Larsen, A. Woods, N. Marsaglia, A. Biswas, S. Dutta, C. Harrison, H. Childs, A flexible system for in situ triggers, in: Proceedings of the Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization, ISAV'18, Association for Computing Machinery, New York, NY, USA, 2018, p. 1–6. doi:10.1145/3281464.3281468.

[24] A. Chaudhuri, T.-H. Wei, T.-Y. Lee, H.-W. Shen, T. Peterka, Efficient range distribution query for visualizing scientific data, in: IEEE Pacific Visualization Symposium (PacificVis), 2014, 2014, pp. 201–208. doi:10.1109/PacificVis.2014.60.

[25] T.-Y. Lee, H.-W. Shen, Efficient local statistical analysis via integral histograms with discrete wavelet transform, Visualization and Computer Graphics, IEEE Transactions on 19 (12) (2013) 2693–2702. doi:10.1109/TVCG.2013.152.

[26] L. Gosink, C. Garth, J. Anderson, E. Bethel, K. Joy, An application of multivariate statistical analysis for query-driven visualization, IEEE Trans. on Vis. and Comp. Graphics 17 (3) (2011) 264–275. doi:10.1109/TVCG.2010.80.

[27] M. Mirzargar, R. T. Whitaker, R. M. Kirby, Curve boxplot: Generalization of boxplot for ensembles of curves, IEEE Transactions on Visualization and Computer Graphics 20 (12) (2014) 2654–2663.

[28] R. T. Whitaker, M. Mirzargar, R. M. Kirby, Contour boxplots: A method for characterizing uncertainty in feature sets from simulation ensembles, IEEE Transactions on Visualization and Computer Graphics 19 (12) (2013) 2713–2722.

[29] D. Thompson, J. A. Levine, J. C. Bennett, P. T. Bremer, A. Gyulassy, V. Pascucci, P. P. Pébay, Analysis of large-scale scalar data using hixels, in: Large Data Analysis and Visualization (LDAV), 2011 IEEE Symposium on, 2011, pp. 23–30. doi:10.1109/LDAV.2011.6092313.

[30] T.-H. Wei, C.-M. Chen, A. Biswas, Efficient local histogram searching via bitmap indexing, Computer Graphics Forum 34 (3) (2015) 81–90. doi:10.1111/cgf.12620.

[31] Y. Wang, W. Chen, J. Zhang, T. Dong, G. Shan, X. Chi, Efficient volume exploration using the gaussian mixture model, IEEE Trans. on Vis. and Comp. Graphics 17 (11) (2011) 1560–1573.

[32] S. Liu, J. Levine, P. Bremer, V. Pascucci, Gaussian mixture model based volume visualization, in: 2012 IEEE Symposium on Large Data Analysis and Visualization (LDAV), 2012, pp. 73–77. doi:10.1109/LDAV.2012.6378978.

[33] D. Silver, X. Wang, Tracking scalar features in unstructured data sets, Proceedings Visualization '98 (Cat. No.98CB36276) 98 (1998). doi:10.1109/VISUAL.1998.745288.

[34] R. Samtaney, D. Silver, N. Zabusky, J. Cao, Visualizing features and tracking their evolution, Computer 27 (1994) 20–27. doi:10.1109/2.299407.

[35] G. Ji, H.-W. Shen, Feature tracking using earth mover's distance and global optimization," pacific graphics 2006.

[36] S. Ozer, J. Wei, D. Silver, K.-L. Ma, P. Martin, Group dynamics in scientific visualization, in: Large Data Analysis and Visualization (LDAV), 2012 IEEE Symposium on, 2012, pp. 97–104. doi:10.1109/LDAV.2012.6378982.

[37] H. Saikia, T. Weinkauf, Global feature tracking and similarity estimation in time-dependent scalar fields, Computer Graphics Forum 36 (3) (2017) 1–11. doi:10.1111/cgf.13163.

[38] F. Sauer, H. Yu, K.-L. Ma, Trajectory-based flow feature tracking in joint particle/volume datasets, IEEE Transactions on Visualization and Computer Graphics 99 (PrePrints) (2014) 1. doi:10.1109/TVCG.2014.2346423.

[39] F. H. Post, B. Vrolijk, H. Hauser, R. S. Laramee, H. Doleisch, The state of the art in flow visualisation: Feature extraction and tracking, Comput. Graph. Forum 22 (4) (2003) 775–792.

[40] Optimizing a new technology to reduce power plant carbon dioxide emissions, https://www.exascaleproject.org/optimizing-a-new-technology-to-reduce-power-plant-carbon-dioxide-emissions/ (2022 (accessed March 23, 2022)).

[41] A. Abad, J. Adánez, A. Cuadrat, F. García-Labiano, P. Gayán, L. F. [de Diego], Kinetics of redox reactions of ilmenite for chemical-looping combustion, Chemical Engineering Science 66 (4) (2011) 689 – 702. doi:https://doi.org/10.1016/j.ces.2010.11.010.

[42] MFiX: Multiphase flows with interphase exchanges, https://mfix.netl.doe.gov (2022 (accessed March 23, 2022)).
URL https://mfix.netl.doe.gov

[43] AMReX: A software framework for massively parallel, block-structured adaptive mesh refinement (AMR) applications, https://amrex-codes.github.io/amrex/index.html (2022

(accessed March 23, 2022)).
URL https://amrex-codes.github.io/amrex/index.html

[44] W. Zhang, A. Myers, K. Gott, A. Almgren, J. Bell, Amrex: Block-structured adaptive mesh refinement for multiphysics applications, The International Journal of High Performance Computing Applications 35 (6) (2021) 508–526. doi:10.1177/10943420211022811.

[45] U. Ayachit, The ParaView Guide: A Parallel Visualization Application, 4th Edition, Kitware Inc., 2015, ISBN 978-1-930934-30-6.

[46] C. Boyce, A. Penn, M. Lehnert, K. Pruessmann, C. Müller, Magnetic resonance imaging of single bubbles injected into incipiently fluidized beds, Chemical Engineering Science 200 (2019) 147 – 166. doi:https://doi.org/10.1016/j.ces.2019.01.047.

[47] K. Buchheit, C. Altantzis, A. Bakshi, T. Jordan, D. Van Essendelft, The bubbletree toolset: Cfd-integrated algorithm for lagrangian tracking and rigorous statistical analysis of bubble motion and gas fluxes for application to 3d fluidized bed simulations, Powder Technology 338 (2018) 960 – 974. doi:https://doi.org/10.1016/j.powtec.2018.07.053.

[48] D. Comaniciu, V. Ramesh, P. Meer, Real-time tracking of non-rigid objects using mean shift, in: Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on, Vol. 2, 2000, pp. 142–149 vol.2. doi:10.1109/CVPR.2000.854761.

[49] A. Elgammal, R. Duraiswami, L. Davis, Probabilistic tracking in joint feature-spatial spaces, in: Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on, Vol. 1, 2003, pp. I–781–I–788 vol.1. doi:10.1109/CVPR.2003.1211432.

[50] H. Obermaier, K. I. Joy, Local data models for probabilistic transfer function design, in: Eurographics Conference on Visualization (EuroVis 2013) Short Papers, 2013, pp. 43–47.

[51] T. Peterka, H. Croubois, N. Li, S. Rangel, F. Cappello, Self-Adaptive Density Estimation of Particle Data, SIAM Journal on Scientific Computing SISC Special Edition on CSE'15: Software and Big Data (2016).

[52] A. Biswas, J. P. Ahrens, S. Dutta, J. M. Musser, A. S. Almgren, T. L. Turton, Feature analysis, tracking, and data reduction: An application to multiphase reactor simulation MFiX-Exa for in-situ use case, Computing in Science Engineering 23 (1) (2021) 75–82. doi:10.1109/MCSE.2020.3016927.

[53] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Süsstrunk, SLIC superpixels compared to state-of-the-art superpixel methods, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (11) (2012) 2274–2282. doi:10.1109/TPAMI.2012.120.

[54] J. Xie, F. Sauer, K.-L. Ma, Fast uncertainty-driven large-scale volume feature extraction on desktop pcs, in: Large Data Analysis and Visualization (LDAV), 2015 IEEE 5th Symposium on, 2015, pp. 17–24. doi:10.1109/LDAV.2015.7348067.

[55] G. Nagino, M. Shozakai, Distance measure between gaussian distributions for discriminating speaking styles, in: Proc. of INTERSPEECH-2006, Vol. 2, 2006, pp. 657–660.

[56] CinemaScience, https://cinemascience.github.io (2022 (accessed March 23, 2022)).
URL https://cinemascience.github.io

[57] W. Schroeder, K. Martin, B. Lorensen, The Visualization Toolkit: An Object Oriented Approach to 3D Graphics, 4th Edition, Kitware Inc., 2004, iSBN 1-930934-19-X.

[58] R. M. Davies, S. G. Taylor, The mechanics of large bubbles rising through extended liquids and through liquids in tubes, in: Proc. R. Soc. Lond., Vol. A200, 1950, p. 375–390.

[59] J. Fang, I. A. Bolotnov, Bubble tracking analysis of pwr two-phase flow simulations based on the level set method, Nuclear Engineering and Design 323 (2017) 68–77. doi:https://doi.org/10.1016/j.nucengdes.2017.07.034.
URL https://www.sciencedirect.com/science/article/pii/S0029549317303667

[60] J. Musser, A. S. Almgren, W. D. Fullmer, O. Antepara, J. B. Bell, J. Blaschke, K. Gott, A. Myers, R. Porcu, D. Rangarajan, M. Rosso, W. Zhang, M. Syamlal, MFIX-Exa: A path toward exascale CFD-DEM simulations, International Journal of High Performance Computing Applications (2021). doi:https://doi.org/10.1177/10943420211009293.

[61] Summit supercomputer, https://docs.olcf.ornl.gov/systems/summit_user_guide.html.