IEEE International Conference on Big Data (IEEEBigData)
December 15-18, 2021

In Situ Adaptive Spatio-Temporal Data Summarization

Soumya Dutta
Los Alamos National Laboratory
Los Alamos, USA
sdutta@lanl.gov

Humayra Tasnim
University of New Mexico
Albuquerque, USA
htasnim30@unm.edu

Terece L. Turton
Los Alamos National Laboratory
Los Alamos, USA
tlturton@lanl.gov

James Ahrens
Los Alamos National Laboratory
Los Alamos, USA
ahrens@lanl.gov

***Abstract*—Scientists nowadays use data sets generated from large-scale scientific computational simulations to understand the intricate details of various physical phenomena. These simulations produce large volumes of data at a rapid pace, containing thousands of time steps so that the spatio-temporal dynamics of the modeled phenomenon and its associated features can be captured with sufficient detail. Storing all the time steps into disks to perform traditional offline analysis will soon become prohibitive as the gap between the data generation speed and disk I/O speed continues to increase. *In situ* analysis, i.e., in-place analysis of data when it is being produced, has emerged as a solution to this problem. In this work, we present an information-theoretic approach for *in situ* reduction of large-scale time-varying data sets via a combination of key and fused time steps. We show that this approach can greatly minimize the output data storage footprint while preserving the temporal evolution of data features. A detailed *in situ* application study is carried out to demonstrate the *in situ* viability of our technique for efficiently summarizing thousands of time steps generated from a large-scale real-life computational simulation code.**

***Index Terms*—Time-varying data, data fusion, in situ analysis, information theory, visualization.**

I. Introduction

With the increase in computing capabilities, large-scale scientific simulations now produce very large data sets containing thousands of time steps. These computer simulations help scientists in understanding the intricate nature of various physical phenomena. All of these phenomena are time-varying in nature and their simulations produce data sets that can take terabytes (TBs) to petabytes (PBs) of disk storage. Storing all such data will be prohibitive since the data generation velocity will outpace the rate at which it can be stored into disks [1], [2]. The bottleneck of slow disk I/O and extreme data volume will entail novel data triage strategies that can work in real-time with the simulation, i.e., *in situ*, and produce informative data summaries, significantly smaller than the raw simulation output, enabling flexible *post hoc* analysis.

Currently, to manage the output data size, simulation scientists often skip regular intervals of time steps and store every nth (n typically varies between 50 $\sim$ 100) time step. By doing so, the scientists remain oblivious of the events that take place in those skipped time steps. A better strategy could be to detect the key time steps and store only the key time steps so that the important events can be preserved. In this case, even though the key time steps are stored, a comprehensive summary of all the time steps will still be missing. Another complicating factor is that many existing key time step detection techniques for scientific data sets assume the availability of all the time steps [3], [4]. For an *in situ* approach, where data becomes available in a streaming fashion, one-time step at a time, such algorithms (a) may not be readily applicable, (b) could be computationally expensive. So, in recent years, researchers have focused on developing *in situ* techniques that allow identification of important time points during the simulation [5]–[7]. However, such techniques typically do not offer any integrated data summarization strategy. Therefore, new automatic *in situ* time-varying data summarization techniques are needed that will produce informative and comprehensive data summaries with minimal storage footprints.

In this work, we propose a spatio-temporal data summarization technique that uses information-theoretic measures to quantify data value importance between consecutive time steps and summarizes data from a sequence of time steps into a single fused data set. As the simulation runs for long hours in supercomputers, the proposed technique analyzes data *in situ*, identifies key time steps based on a user-provided criterion, and summarizes the data between every two consecutive key time steps into a single summarized data set that captures a comprehensive view of the features for the time window. The proposed method stores raw simulation data for each key time step along with time-varying data summaries for time steps between every two key time steps. We show that the output data size for our method is significantly smaller compared to the raw simulation data size and that the summary data can be visualized interactively during *post hoc* exploration. To show the efficacy of the proposed technique, we conduct a detailed *in situ* application study with a large-scale simulation. Therefore, our contributions in this work are twofold:

- We propose an adaptive spatio-temporal data summarization technique for large-scale time-varying data sets that produces summary data as a combination of key and fused time steps to preserve: (a) the important events, and (b) a comprehensive view of the simulation data.

- We study the effectiveness of the proposed algorithm *in situ* with a large-scale simulation and demonstrate its practical applicability and *in situ* viability.

II. Related Works

With modern supercomputers producing large-scale data sets, *in situ* analysis has emerged as a promising solution and several *in situ* analysis frameworks such as Ascent [8], ParaView Catalyst [9], and VisIt libSIM [10] have been developed. Further, a significant amount of research has been done to develop data reduction techniques for producing reduced data summaries that can be stored and used as a proxy for the raw data. Cinema [11] is such an *in situ* image-based data reduction and visualization approach. Among other *in situ* techniques, compression [12], sub-sampling [13], [14], and distribution-based summaries [1], [15] are popular. In this work, we advocate a hybrid approach where we store the raw data for important key time steps and summarize the intermediate time steps to achieve sufficient data reduction.

Detection of key time points in a data set is an important problem for time-varying data analysis. Several approaches have been proposed for key time step detection for large time-varying data sets [3], [6], [16]. These techniques generally allow the detection of key time points and do not offer any data summarization capability. The computer vision community has developed several techniques for doing spatio-temporal fusion of large data obtained from different sources. Pulong and Kang proposed a technique for data fusion [17]. Nguyen et al. [18] developed a technique for summarizing large spatio-temporal images. In a recent work, Shah et al. [19] proposed an algorithm for real-time summarization of data streams for smart grid applications.

The use of information-theoretic measures [20], [21] to solve data analysis and visualization problems is well-known. Mutual information has been used to perform data registration [22], [23], view selection [24], and for quantifying information transfer from data to image space [25]. Various decomposition of mutual information, called specific mutual information and pointwise mutual information measures have become recently popular for fusing multi-modal data [26] and multivariate sampling [27] for data reduction. For a detailed review of information theory applications in data analysis and visualization, interested readers are referred to [28], [29].

III. Method

In this work, we propose a new technique for summarizing a sequence of time-varying scalar fields into a single scalar field that captures the dynamic temporal evolution of the data features. The users can study the summary fields to obtain a comprehensive view of the time-varying nature of the features. This approach achieves significant data reduction for the *post hoc* analysis while preserving the important feature dynamics of a sequence of time steps. Note that we develop this algorithm for *in situ* use cases where we run our algorithm online when the simulation is running and access the time step data one by one in a streaming fashion.

A. Data Value Informativeness Quantification

Since the goal is to combine data from a sequence of time steps, it is important to quantify the informativeness of each data point so that we can prioritize one data point over others during the summarization process. In information theory [20], mutual information (MI) is a well-known measure that estimates the amount of information overlap between two random variables and can be formally computed following Equation 1:

$$I(Y;X) = \sum_{y\in Y}\sum_{x\in X} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \tag{1}$$

In Equation 1, $I(Y;X)$ is the MI between two random variables Y and X, $y \in Y$ represents a specific value of Y and $x \in X$ is a value of X. The joint probability between x and y is written as $p(x,y)$ and the marginal probabilities of x and y are $p(x)$ and $p(y)$ respectively. MI for two random variables computes to a single number reflecting the total shared information between X and Y. Since we need information content of each data value so that we can perform spatio-temporal data summarization, we focus on a decomposition of MI that can estimate the information content of each data value of one variable, while observing values from another variable. Such decomposition of MI is called *specific information*.

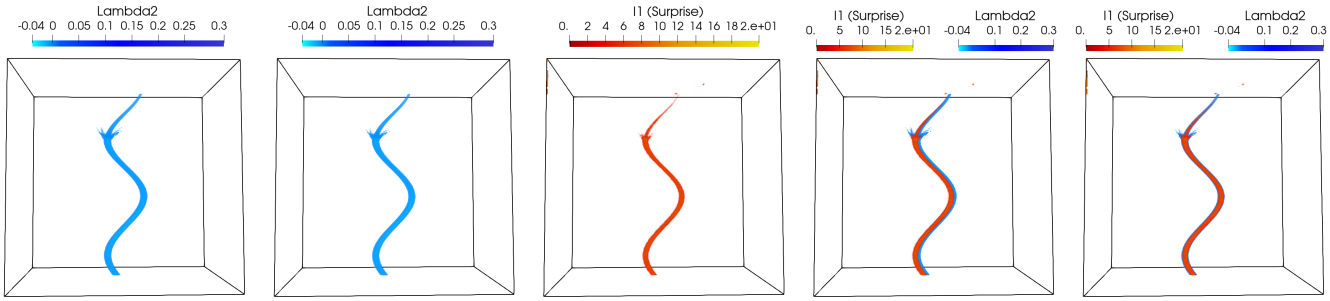
Specific information measure was first introduced by DeWeese and Meister [30] and can be formally derived from Equation 1 as shown in Equation 2 and 3. The specific information, called *surprise*, denoted as $I_1(y;X)$ in Equation 3, represents the informativeness of a data value y when the whole variable X is observed. Here, $p(x|y)$ represents the conditional probability of value x given y.

$$\begin{aligned} I(Y;X) &= \sum_{y\in Y} p(x)\sum_{x\in X} p(x|y)\log\frac{p(x|y)}{p(x)} \\ &= \sum_{x\in X} p(x) I_1(y;X), \end{aligned} \tag{2}$$

$$I_1(y;X) = \sum_{x\in X} p(x|y)\log\frac{p(x|y)}{p(x)} \tag{3}$$

For a data value y, a high value of $I_1(y;X)$ indicates that some infrequent occurrences of $x \in X$ have become more probable after observing the value y from Y, amounting to a surprising result, hence the name surprise. The value of surprise ($I_1(y;X)$) is always positive, i.e., $I_1(y;X) \geq 0 \quad \forall \quad y \in Y$ since it represents the KL-divergence between the distributions $p(X)$ and $p(X|y)$ [30].

We use surprise as the measure to estimate the informativeness of a data value when data values from another time step are observed. More specifically, if we assume that X and Y represent the same data variable from time step t and $t+1$,



(a) T=25

(b) T=26

(c) $I1field$ generated using tornado data at T=25 and 26.

(d) $I1field$ overlapped with tornado data at T=25

(e) $I1field$ overlapped with tornado data at T=26

Fig. 1. Visualization of $I1field$ generated using two consecutive time steps of the analytical Tornado data set. Volume rendering technique is used to generate the visualization results. Figure 1(a) and 1(a) show the vortex region of the Tornado data and Figure 1(c) shows the corresponding $I1field$. In this illustrative example, data from T=25 is observed and so the high I_1 valued region overlap accurately with the vortex region at T=26 as shown in Figure 1(e).

then we can estimate the informativeness of each data value at time step $t+1$ as $I_1(y;X)$, by observing the same variable from the previous time step t. This gives us a way of finding the highly surprising regions in the data when we compare it with a previous time step. These surprising regions (i.e., regions with high $I_1(y;X)$ values) can indicate the regions where the data features exist.

B. Information Fields

Our primary target application is time-varying 3-D scalar fields with the goal to summarize a sequence of 3-D scalar fields into a single scalar field that can provide a comprehensive summary of the data features for the selected time sequence. Such summaries can indicate how the data features of interest have evolved within the time window and can also reveal their tracking information. Equation 3 shows how surprise can be estimated for each data value in variable Y. In practice, computation of such information theory measures is done by first establishing a communication channel $Y \rightarrow X$ between the variables X and Y as discussed in [26] and then computing the surprise using the communication channel. Normalized histograms can be used to estimate probability distributions while computing the values of $I_1(y;X)$. After the surprise (I_1) values are computed, we create a new scalar field where at each spatial grid point (with data value $y \in Y$), we put the corresponding value of $I_1(y;X)$. Since such a scalar field contains information values at each grid point, it can be called an *information field* or $I_1 field$. The $I_1 field$ computed between two time steps can be visualized directly and regions with high I_1 values can be explored as salient regions.

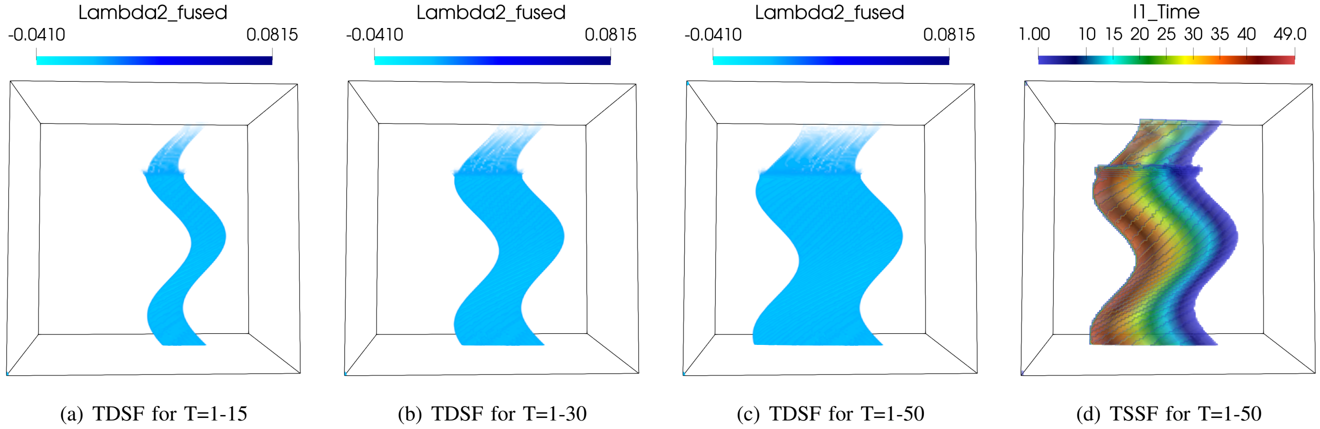
Figure 1 shows an example of an $I_1 field$ constructed using two time steps of an analytical Tornado data set. This data set of dimension $128 \times 128 \times 128$, contains velocity vectors and is generated by an analytical function [31]. The data set has 50 time steps and simulates a tornado-like vortex structure. For this study, we have modified the analytical equation so that the center of the tornado changes position with time, creating a moving vortex in the spatial domain. Tracking and visualizing this vortex is of interest in this data. To detect the vortex region, we have used the lambda2 (λ_2) vortex criterion [32]. The visualizations shown here are generated using the Ray-casting-based Volume Rendering technique [33] from ParaView [34] that allows interactive visualization of 3-D scalar field data sets. Figure 1(a) and 1(b) show the vortex at T=25 and T=26 respectively. Even though they look very similar, the vortex at T=26 has moved slightly toward the left from its position at T=25. Figure 1(c), presents the $I_1 field$ computed at T=26 when the data at T=25 is observed. We refer to the time step that is the observed variable as the reference time step. We find that the $I_1 field$ at T=26 captures the location of the vortex region accurately. In Figure 1(d) and 1(e), we superimpose the estimated $I_1 field$ with the λ_2 vortex fields from T=25 and T=26 respectively. Figure 1(d) shows that the $I_1 field$ at T=26 captures the slight shift on the vortex structure and only partially overlaps with the vortex at T=25, whereas, in Figure 1(e), a complete overlap of the $I_1 field$ with the underlying vortex is seen at T=26.

C. Time-varying Feature-based Data Summarization using Information Fields

The insights obtained from Figure 1 allow us to develop the idea of time-varying data fusion using $I_1 fields$ from a sequence of consecutive time steps. One can imagine that if we compute the $I_1 fields$ for every consecutive pair of time steps, each $I_1 field$ will assign high values to the statistically salient regions of the data. Then, if we create a new fused summary field where at each spatial location, we assign the data value from the time step where the I_1 value is the highest over the chosen time window, we can combine all the high I_1 valued regions from a time window into a single field. Hence, for each spatial location p, the assigned value is calculated as:

$$Val(p) = max(I_1^t(p)), \forall\, t = t_{start}, .., t_{end} \tag{4}$$

where t_{start} and t_{end} represent start and end time steps, $I_1^t(p)$ is the value of I_1 at point p in time t. Conceptually, this technique will maximize the spatio-temporal information in the combined field by selecting data points that have maximum I_1 values over the time window. This combined field will



(a) TDSF for T=1-15

(b) TDSF for T=1-30

(c) TDSF for T=1-50

(d) TSSF for T=1-50

Fig. 2. Demonstration of the proposed spatio-temporal data summarization scheme using a sequence of time steps from the Tornado data set. Figure 2(a), 2(b), and 2(c) show the TDSFs of the Tornado data when time steps between 1-15, 1-30, and 1-50 have been summarized using the proposed algorithm. In Figure 2(d) we present the TSSF for the Tornado data corresponding to the TDSF shown in Figure 2(c). The colors in Figure 2(d) shows the temporal evolution of the vortex region over the time window and how it moves gradually from right to left.

capture the time-varying pattern of the data by focusing on the salient regions with high I_1 values.

Since domain scientists primarily want to study the important features in their data, we devise our summarization strategy for the feature regions when a domain-specific feature descriptor is available. This methodology allows the user to provide a feature descriptor, such as a threshold, and while performing the temporal summary, we check if the current data point is a feature and then only summarize such points. For all the non-feature points in the data, we assign a constant value to them so that when the summary fields are analyzed and visualized, the non-feature points can be emphasized less using volume rendering techniques so that the users can focus on the evolution of the features without any occlusion from non-featured regions.

Figure 2 demonstrates this spatio-temporal data summarization scheme using the analytical Tornado data. Figure 2(a), 2(b), and 2(c) show the volume rendering of the summary fields when 15, 30, and 50 time steps of Tornado data are summarized into a single field. These summary fields are denoted as the *temporal data summary field* (TDSF). It is seen that these TDSFs can capture the evolution of the vortex in Tornado data as the vortex moves from right to left. To capture how the TDSFs are generated and associate each part of the TDSF with its relevant time step, we also generate another field, the *time step summary field* (TSSF). For each spatial location, the TSSF assigns the time step number from which the data (with the highest I_1 value) is selected. Figure 2(d) shows the TSSF for Tornado data that corresponds to TDSF at Figure 2(c). The colors in Figure 2(d) reflect the time steps and, using a colormap that naturally delineates bands, we can see that the vortex moves from right to left over time as the color changes from blue to red.

By exploring the TDSF and TSSF together, users can get a comprehensive view of the evolution of the vortex in the Tornado data without needing to inspect each time step individually. Disk storage can be significantly reduced by retaining the raw Tornado data at T=1 (initial time step) and T=50 (final time step) while keeping the TDSF and TSSF fields as a replacement for all the 48 intermediate time steps. We observe that the storage for the raw Tornado data is 489MB, whereas the proposed technique will only take 40MB disk space, achieving approximately 92% storage reduction. Using this technique, we can generate temporal summary fields (TDSFs and TSSFs) for sequences of time steps, retaining raw simulation data for the start and end time steps of each sequence along with the corresponding TDSFs and TSSFs for achieving sufficient data reduction.

IV. In Situ Application Study

A. Application Background

In this section, we apply our algorithm *in situ* to a data set generated from a large-scale computational fluid dynamics code, MFIX-Exa [35], [36], which is being developed at the National Energy Technology Laboratory (NETL). MFIX-Exa generates particle-based data to study the working principles of chemical looping reactors (CLR). Such reactors contain fluidized beds where particles interact and, under certain physical conditions, bubbles (void regions) are formed. The study of the dynamics and interaction of such bubbles is critical since the formation of large, fast-moving bubbles in fluidized beds can cause poor gas/solid mixing, lowering the conversion efficiency and stability of the reactor. Data produced from an MFIX-Exa run can contain millions of particles per time step and thousands of time steps, needing terabytes to petabytes of storage. As a consequence, storing all the raw particle data for a *post hoc* analysis will be prohibitive. To address this need, we have deployed our proposed algorithm *in situ* and generated bubble-based summarization fields so that the raw particle data are not required to be stored at each time step, thereby significantly reducing the overall storage needs.

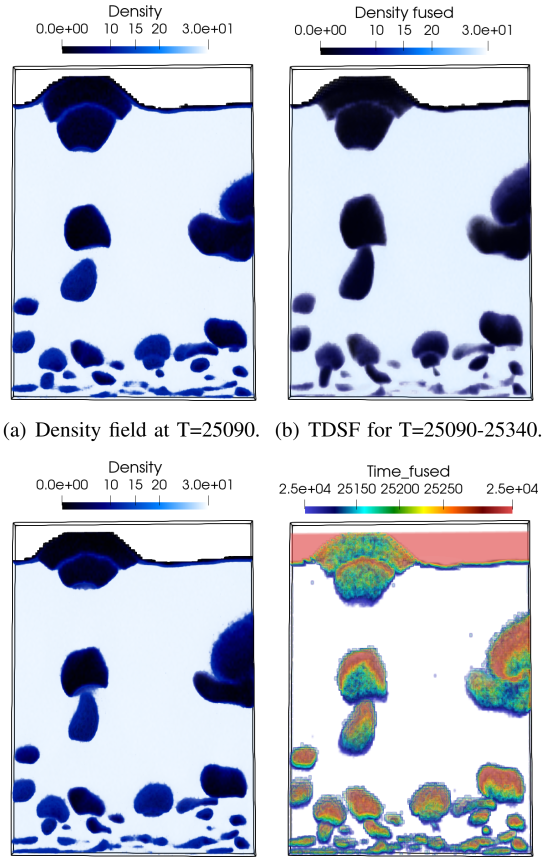
To perform *in situ* analysis using MFIX-Exa, custom code is added to MFIX-Exa code bases. Our *in situ* code is developed in C++ and uses the VTK [37] library for data processing. We have developed an *in situ* adapter function that directly accesses the raw particle data. As the simulation code and our *in situ* algorithm run on the same memory and computing resources, this *in situ* integration works in synchronous mode, tightly coupled with the simulation code.

B. *In Situ Algorithm for Streaming Data*

Since MFIX-Exa produces particle fields, we first convert it to a scalar particle density field. To estimate the particle density, we create a spatial 3-D histogram using particle locations. As the bin frequencies of this 3-D histogram reflect the number of particles in a local region of the domain, we convert this 3-D spatial histogram into a regularly structured grid data where the number of histogram bins translates to the spatial dimensions of the structured grid and the bin frequency values are interpreted as particle density at each grid point.

A threshold on these particle density fields can be used to segment the bubbles. The MFIX-Exa domain scientists want to understand these complex bubble interactions while evaluating their computational model. The interesting time points for this simulation are when relatively larger bubbles undergo a merge/split event. However, since the simulation data gradually evolves over time and such events do not happen at each time step, this is an ideal use case for our approach. In this case, the sequence of time steps between merge/split events can be summarized into a fused field. To preserve the raw particle data at the key time steps when a merge/split event happens, we first segment the density field and count the number of segments where each segment indicates a bubble. For the next time step, if the number of segments remains the same as the previous time step – indicating no merge/split has happened – we apply our summarization algorithm to fuse all such intermediate time steps. When the count of the bubbles changes, the algorithm outputs the summarized TDSF and TSSF at that time step and also stores the raw particle data, re-initializes the TDSF and TSSF, and continues the process from the next time step.

The algorithm uses a threshold value to segment and detects the bubble regions (regions $\leq$ TH) while generating the summarized fields. In the *in situ* environment, we only have access to one-time step at a time, requiring modifications to the methodology. Since the size of the estimated particle density field is quite small, we keep the particle density field from the previous time step in memory. The joint histogram computation needed to compute the surprise (I_1) values requires two sequential time steps. We also initialize TDSF and TTSF as global data objects. At each new time step, for every spatial location, if the value of I_1 is higher than the current value, we update the data value at that location with the data value from the current time step and also update the time step number with the current time step number for the same spatial location in the TSSF. This process incrementally constructs the TDSF and TSSF for a sequence of time steps in the *in situ* setting. Once the bubble count changes, we output the current TDFS, TSSF, and the particle raw data and reinitialize the TDSF and TSSF using the values from the current time step. This algorithm can run continuously with the simulation and produce TDSFs and TSSFs for sequences of time steps when the bubble count remains the same. Hence, this method adaptively stores key time steps from the MFIX-Exa simulation and summarizes the intermediate time steps.



(a) Density field at T=25090. (b) TDSF for T=25090-25340.

(c) Density field at T=25340. (d) TSSF for T=25090-25340.

Fig. 3. *In situ* application study results of the proposed method when run with the MFIX-Exa simulation. Figure 3(a), and 3(c) show the particle density field from T=25090 and 25340 respectively. The bubble features are observed with dark blue regions. The TDSF generated for the intermediate time steps (T=25090-25340) is provided in Figure 3(b) and the corresponding TSSF is shown in Figure 3(d). We find that the TSSF is able to provide a comprehensive summary of all the bubbles within this time window and as two bubbles (two bubbles at the center left of Figure 3(a)) merge at T=25340, the bubble count changes at T=25340 and the proposed technique outputs summary results.

C. *Analysis Results*

We have tested the effectiveness of our method by running it *in situ* with the MFIX-Exa simulation. The simulation test case represents a scenario where a constant density, constant viscosity gas is used to fluidize spherical particles of uniform radius. The fluidized bed has a constant velocity gas inlet at the bottom of the bed and the simulation contains approximately 4.1 million particles. We have run this simulation for 6000 time steps starting from a previously stored checkpoint file at T=25000 to reach the point when bubbles are already forming.

In Figure 3, we show the results of *in situ* data summarization for one of the time windows, with start time step 25090 and end time step 25340. Figure 3(a) and 3(c) show the estimated density fields at T=25090 and T=25340 respectively. We observe that the bubbles (dark regions with low particle density) have evolved. To focus the analysis only on the bubble regions, we have used the density threshold=12 for segmenting the bubbles. Also, since the state of the bubbles changes very slowly between two consecutive time steps, we call the *in situ* routine at every 10^{th} time step. Furthermore, since the domain experts are more interested in the evolution of larger bubbles, in this study, we only count the number of bubbles containing more than 750 connected grid points. In Figure 3(b) and Figure 3(d), we present the TDSF and TSSF for this window that show the evolution of the bubbles for this intermediate time steps. Note that at T=25340, two bubbles merge (the bubbles at the center-left of Figure 3(a)) and as a result, the number of bubbles changes. To preserve this time step as one of the key time steps, our technique outputs the raw particle data along with the summarized TDSF and TSSF for the time window T=25090-25340. For the entire *in situ* run of 6000 time steps, our method identified 54 key time points, summarizing the intermediate data for each pair of consecutive key time steps between key time points. These results demonstrate the usefulness of the proposed method for analyzing and summarizing large-data sets *in situ* where we can access the simulation data at a much higher temporal frequency, bypassing the expensive disk I/O.

D. Storage Savings and Computational Performance

The *in situ* studies were done in the cluster Cori at the National Energy Research Scientific Computing Center (NERSC). NERSC is one of the primary high-performance scientific computing facilities for the Office of Science in the U.S. Department of Energy. Cori is a Cray $XC40$ system, capable of achieving a peak performance of about 30 petaflops.

For this study, the raw particle data is stored using PLOT FILE format containing particle ids, particle locations, and their velocities. We ran 6000 time steps of the simulation. The proposed method stored 54 key time steps with the TDSFs and TSSFs. The spatial dimension of the generated TDSFs and TSSFs are $128 \times 16 \times 128$ and are stored in VTK format. We find that the proposed method needs 16.03 GB storage, while if we store all the raw data for every 10^{th} time step, then we would require 170 GB storage. Hence the proposed method is able to reduce approximately 91% disk storage.

In Table I, we provide the *in situ* computational performance of our technique. Typically, when an *in situ* analysis is performed, it is desirable that the *in situ* processing takes only a small additional fraction of the simulation time. Our study is run using 1024 processors and it is observed that the *in situ* processing time is significantly smaller compared to the simulation time. Also, from the fifth and sixth column of Table I, we observe that the *in situ* I/O, which includes timings for storing the raw data for key time steps and the TDSFs and TSSFs, is significantly smaller compared to the raw data I/O if we store the particle data at every 10^{th} time step to conduct similar analysis offline. In addition, we also measure the timings if our algorithm is executed *post hoc* and found that the *post hoc* disk I/O takes 246.27 minutes, which is significantly higher compared to the *in situ* I/O. However, by processing the data *in situ*, we are able to bypass this slow *post hoc* I/O. Therefore, by performing *in situ* analysis, the proposed method saves both the storage and computational time while enabling flexible *post hoc* analysis.

TABLE I
COMPUTATIONAL PERFORMANCE FOR THE IN SITU APPLICATION STUDY USING MFIX-EXA SIMULATION.

	No. of processors	Simulation (mins)	In situ processing (mins)	Simulation raw I/O (mins)	In situ I/O (mins)
MFIX-Exa Case (∼4.1M particles, 6000 time steps)	1024	553.05	24.37	72.7	2.26

V. CONCLUSION & FUTURE WORKS

In conclusion, we have presented an *in situ* technique for summarizing large-scale spatio-temporal data sets to reduce the size of the output data significantly while preserving the important state of the features. The proposed method detects key time steps based on a suitable user-provided criterion and fuses data between every pair of key time steps into a summarized data set. Finally, the summary data sets are stored along with the raw data from the key time steps so that they can be analyzed and visualized during *post hoc* exploration. We verify the efficacy of our method by conducting an *in situ* study with a large-scale simulation.

In the future, we plan to develop criteria for detecting key time steps that will not need any domain knowledge so that key time steps can be detected in a purely data-driven way which will make the algorithm applicable across a wide range of scientific data sets. We also wish to run a GPU implementation of this technique with a larger case of MFIX-Exa to study the computational performance further.

ACKNOWLEDGMENT

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-21-28689 Ver. 2. This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory, operated under Contract No. DE-AC02-05CH11231. We thank our ECP collaborators, especially J. Musser and A. Almgren.

REFERENCES

[1] S. Dutta, C. Chen, G. Heinlein, H.-W. Shen, and J. Chen, "In situ distribution guided analysis and visualization of transonic jet engine simulations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 811–820, 2017.

[2] H. Childs, "Data exploration at the exascale," *Supercomputing frontiers and innovations*, vol. 2, no. 3, 2015. [Online]. Available: http://superfri.org/superfri/article/view/78

[3] X. Tong, T.-Y. Lee, and H.-W. Shen, "Salient time steps selection from large scale time-varying data sets with dynamic time warping," in *IEEE Symposium on Large Data Analysis and Visualization (LDAV)*, 2012, pp. 49–56.

[4] C. Wang, H. Yu, and K.-L. Ma, "Importance-driven time-varying data visualization," *IEEE Trans. on Vis. and Comp. Graphics*, vol. 14, no. 6, pp. 1547–1554, 2008.

[5] M. Salloum, J. C. Bennett, A. Pinar, A. Bhagatwala, and J. H. Chen, "Enabling adaptive scientific workflows via trigger detection," in *Proceedings of the First Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization*, ser. ISAV2015. New York, NY, USA: Association for Computing Machinery, 2015, p. 41–45. [Online]. Available: https://doi.org/10.1145/2828612.2828619

[6] K. Myers, E. Lawrence, M. Fugate, C. M. Bowen, L. Ticknor, J. Woodring, J. Wendelberger, and J. Ahrens, "Partitioning a large simulation as it runs," *Technometrics*, vol. 58, no. 3, pp. 329–340, 2016.

[7] M. Larsen, A. Woods, N. Marsaglia, A. Biswas, S. Dutta, C. Harrison, and 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*, ser. ISAV'18. New York, NY, USA: Association for Computing Machinery, 2018, p. 1–6. [Online]. Available: https://doi.org/10.1145/3281464.3281468

[8] M. Larsen, J. Ahrens, U. Ayachit, E. Brugger, H. Childs, B. Geveci, and 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*, ser. ISAV'17. New York, NY, USA: Association for Computing Machinery, 2017, p. 42–46. [Online]. Available: https://doi.org/10.1145/3144769.3144778

[9] N. Fabian, K. Moreland, D. Thompson, A. C. Bauer, P. Marion, B. Gevecik, M. Rasquin, and 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.

[10] B. Whitlock, J. M. Favre, and 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*, ser. EGPGV '11. Eurographics Association, 2011, pp. 101–109.

[11] J. Ahrens, S. Jourdain, P. OLeary, J. Patchett, D. H. Rogers, and 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.

[12] H. Lehmann and 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.

[13] J. Woodring, J. Ahrens, J. Figg, J. Wendelberger, S. Habib, and 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.

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

[15] Y. C. Ye, T. Neuroth, F. Sauer, K.-L. Ma, G. Borghesi, A. Konduri, H. Kolla, and 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.

[16] B. Zhou and Y.-J. Chiang, "Key time steps selection for large-scale time-varying volume datasets using an information-theoretic storyboard," *Computer Graphics Forum*, vol. 37, no. 3, pp. 37–49, 2018.

[17] P. Ma and E. L. Kang, "Spatio-temporal data fusion for massive sea surface temperature data from modis and amsr-e instruments," *Environmetrics*, vol. 31, no. 2, p. e2594, 2020.

[18] H. Nguyen, M. Katzfuss, N. Cressie, and A. Braverman, "Spatio-temporal data fusion for very large remote sensing datasets," *Technometrics*, vol. 56, no. 2, pp. 174–185, 2014.

[19] Z. Shah, A. Anwar, A. Mahmood, Z. Tari, and A. Y. Zomaya, "A spatiotemporal data summarization approach for real-time operation of smart grid," *IEEE Transactions on Big Data*, vol. 6, no. 04, pp. 624–637, oct 2020.

[20] T. M. Cover and J. A. Thomas, *Elements of Information Theory 2nd Edition (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006.

[21] S. Verdú, "Fifty years of shannon theory," *Information Theory, IEEE Transactions on*, vol. 44, no. 6, pp. 2057–2078, Oct 1998.

[22] D. L. G. Hill, P. G. Batchelor, M. Holden, and D. J. Hawkes, "Medical image registration," *Physics in Medicine and Biology*, vol. 46, no. 3, p. R1, 2001.

[23] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens, "Multimodality image registration by maximization of mutual information," *Medical Imaging, IEEE Transactions on*, vol. 16, no. 2, pp. 187–198, April 1997.

[24] I. Viola, M. Feixas, M. Sbert, and M. E. Gröller, "Importance-driven focus of attention," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 12, no. 5, pp. 933–940, 2006.

[25] R. Bramon, M. Ruiz, A. Bardera, I. Boada, M. Feixas, and M. Sbert, "An information-theoretic observation channel for volume visualization." *Comput. Graph. Forum*, vol. 32, no. 3, pp. 411–420, 2013.

[26] R. Bramon, I. Boada, A. Bardera, J. Rodriguez, M. Feixas, J. Puig, and M. Sbert, "Multimodal data fusion based on mutual information," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 18, no. 9, pp. 1574 –1587, sept. 2012.

[27] S. Dutta, A. Biswas, and J. Ahrens, "Multivariate pointwise information-driven data sampling and visualization," *Entropy*, vol. 21, no. 7, pp. 1–25, 2019.

[28] M. Chen, M. Feixas, I. Viola, A. Bardera, H.-W. Shen, and M. Sbert, *Information Theory Tools for Visualization*. A K Peters/CRC Press, August 25, 2016.

[29] M. Sbert, M. Feixas, J. Rigau, M. Chover, and I. Viola, *Information Theory Tools for Computer Graphics*, ser. Synthesis Lectures on Computer Graphics and Animation. Morgan and Claypool Publishers Colorado, 2009.

[30] M. R. DeWeese and M. Meister, "How to measure the information gained from one symbol." *Network: Computation in Neural Systems*, no. 4, pp. 325–340, nov 1999.

[31] R. Crawfis and N. Max, "Texture splats for 3d scalar and vector field visualization," in *Visualization, 1993. Visualization '93, Proceedings., IEEE Conference on*, Oct 1993, pp. 261–266.

[32] J. Jeong and F. Hussain, "On the identification of a vortex," *Journal of Fluid Mechanics*, vol. 285, pp. 69–94, 1995.

[33] R. A. Drebin, L. Carpenter, and P. Hanrahan, "Volume rendering," in *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '88. New York, NY, USA: Association for Computing Machinery, 1988, p. 65–74. [Online]. Available: https://doi.org/10.1145/54852.378484

[34] U. Ayachit, *The ParaView Guide: A Parallel Visualization Application*, 4th ed. Kitware Inc., 2015, ISBN 978-1-930934-30-6. [Online]. Available: http://www.paraview.org/paraview-guide/

[35] "MFIX-Exa," https://amrex-codes.github.io/MFIX-Exa/docs_html/, 2021 (accessed August 25, 2021).

[36] 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, and M. Syamlal, "MFIX-Exa: A path toward exascale CFD-DEM simulations," *International Journal of High Performance Computing Applications*, 2021.

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