Homogeneity Guided Probabilistic Data Summaries for Analysis and Visualization of Large-Scale Data Sets

Soumya Dutta[*] The Ohio State University
Jonathan Woodring[†] Los Alamos National Laboratory
Han-Wei Shen[‡] The Ohio State University
Jen-Ping Chen[§] The Ohio State University
James Ahrens[¶] Los Alamos National Laboratory

Abstract

High-resolution simulation data sets provide plethora of information, which needs to be explored by application scientists to gain enhanced understanding about various phenomena. Visual-analytics techniques using raw data sets are often expensive due to the data sets' extreme sizes. But, interactive analysis and visualization is crucial for big data analytics, because scientists can then focus on the important data and make critical decisions quickly. To assist efficient exploration and visualization, we propose a new region-based statistical data summarization scheme. Our method is superior in quality, as compared to the existing statistical summarization techniques, with a more compact representation, reducing the overall storage cost. The quantitative and visual efficacy of our proposed method is demonstrated using several data sets along with an *in situ* application study for an extreme-scale flow simulation.

Index Terms: I.3 [COMPUTER GRAPHICS]: Picture/Image Generation—Display algorithms; G.3 [PROBABILITY AND STATISTICS]: Distribution functions—Statistical computing.

1 Introduction

Recent advancements in high performance computing have enabled application scientists to perform computational simulations with very high-resolution models. Large-scale simulations, now-a-days, generate different types of data output in the order of petabytes and beyond. These simulations allow experts to model various physical phenomena with high precision. Detailed exploration of such data sets can enhance the understanding about the modeled phenomena greatly. However, timely analysis and visualization of such sheer amount of data is posing significant challenges for the scientists.

The use of statistical data summaries has emerged as a promising approach for analyzing and visualizing large-scale data sets [17, 20, 22, 31–34]. The applications of probabilistic data summaries for big data analytics is becoming more and more prominent in the visualization domain [7, 14, 15, 30, 40]. Specifically, local region-based distribution data summaries for feature exploration and tracking in scientific applications have been explored [14, 15, 17, 18, 29, 36, 38]. The benefits of local region-based probabilistic data models are fourfold: (1) Representing a block of data with a probability distribution can preserve the block's statistical properties well, which allow efficient feature analysis [15, 18]; (2) A compact distribution-based data model is able to reduce the size of the data significantly which enables flexible and scalable exploration of extreme-scale data sets [14]; (3) Uncertainty quantification during analysis becomes possible which enriches verifiable visualization [14, 36]; (4) By sampling the distributions, a statistical realization of the raw data can be constructed and visualized for exploration [7, 19, 26].

An ideal local region-based statistical data summarization scheme aims at preserving the statistical properties of the data as much as possible with a compact representation. Therefore, for achieving a compact-yet-accurate probabilistic data representation, a region partitioning scheme produces partitions with coherent data values, such that, efficient distribution-driven summarization is possible, and statistical uncertainty in visual analytics can be reduced. However, a majority of the previous works modeled the data in local regions by estimating distribution over a regular spatial partitioning of the domain. These methods have demonstrated good results, but have several potential shortcomings. A regular partitioning does not consider any inherent spatial data coherency. As a result, many data blocks will have high data value variation resulting lower accuracy in sampling and higher uncertainty during visualization. Furthermore, as regular partitioning does not consider data homogeneity during decomposition, visualization will introduce artifacts and discontinuities on block boundaries, making the visualization less effective. Hence, there is a growing need of more accurate and efficient statistical data summarization techniques, judging by the wide applicability of local statistical data models in visualization community.

In this work, we propose an improved local region-based statistical data summarization technique using distributions. The proposed method partitions data by its spatial coherency and aims to reduce uncertainty of all partitions. To partition the data into local regions, we employ SLIC (Simple Linear Iterative Clustering) algorithm [1], which was used for generating super-pixels and super-voxels [1,46]. This minimizes the variance in each spatial partition and hence, each region/partition can be compactly summarized using a probability distribution function which preserves the statistical properties of the data efficiently. To achieve this, we propose a hybrid scheme of distribution-based summarization by using either a single Gaussian or a mixture of Gaussians (GMM) per partition. Advantages of using GMMs as a compact parametric distribution representation over other alternatives such as histograms and Kernel Density Estimators (KDE) have been discussed previously in [15, 26]. Furthermore, GMMs also have been shown to be effective for probabilistic data classification [14,26,39] which makes it an attractive choice in this work.

For evaluating the efficacy of the proposed technique, we conduct extensive quantitative and qualitative studies among: (a) Our SLIC-based method; (b) Regular partitioning; and (c) K-d tree partitioning. For each of these partitioning methods, the data summarization is done using the aforementioned hybrid summarization scheme. We study two visualization applications in our experimentation for showing effectiveness of our method. The results demonstrate that: (a) Both quantitatively and qualitatively, our SLIC-based summarization produces superior statistical sampling-based data reconstruction with best storage-to-quality trade-off; (b)

[*] e-mail: dutta.33@osu.edu
[†] e-mail:woodring@lanl.gov
[‡] e-mail:shen.94@osu.edu
[§] e-mail:chen.1210@osu.edu
[¶] e-mail:ahrens@lanl.gov

More precise distribution similarity-based feature matching, where identified features are free from boundary discontinuities and other artifacts, which often arise from regular or k-d partitioning scheme. We also show that, our method is suitable for *in situ* summarization, by running the proposed scheme directly with a large-scale CFD simulation, resulting in an improved distribution-based data summarization enabling flexible and scalable post-hoc analysis.

Our contributions in this work are twofold:

1. We propose a novel and improved statistical data summarization technique for large-scale simulation data which enables *in situ* triage and summarization of data while preserving the important information compactly.
2. We present a comprehensive study among the existing partition-based data summarization methods and the proposed scheme to demonstrate the superiority of our proposed scheme, both quantitatively and qualitatively.

2 Related Works

Statistics and distribution-driven data visualization. Statistical analysis methods for data exploration and visualization has numerous applications in visualization community. Use of distribution-based methods for exploring scientific data sets has become an emerging trend in the visualization domain. For visualizing spatial distribution data sets, Kao et al. [21], Luo et al. [30] and Potter et al. [34] visualized distribution datasets by displaying statistical summaries such as means, standard deviations and skews in color, height field, or glyphs. Potter et al. [33] utilized summary plots which enhance box plots with moments and histograms in higher dimension. Kniss et al. proposed statistically salient volume data visualization [22]. A study of Non-parametric distribution models and their applicability was discussed in [32]. A fuzzy matching based feature extraction method was proposed by Johnson and Huang [20]. Efficient range distribution query algorithms using integral histograms [7] and wavelet transforms [24] yielded valuable statistical information from data. Wang et al. [39] utilized GMMs for transfer function design in time-varying datasets. Liu et al. [26] exploited GMMs for stochastic sampling-based volume rendering on the GPU. For analyzing FTLE in distribution data sets, Guo et al. [19] used distribution-based data models for uncertainty quantification.

Local region-based distribution models for large-scale data visualization. For designing transfer functions, Lundstrom et al. [29] used local histograms. Wei et al. [40] presented efficient local histogram search using bitmap indexing for feature analysis. For large data summarization, Thompson et al. [36] made use of distribution-based hixels, which stored a histogram per data block to preserve uncertainty information due to data down-sampling. A regular block-wise approach was taken by Gu and Wang for a graph based analysis of time-varying data [18]. Recently, Dutta and Shen [15] proposed uncertain feature extraction and tracking based on block-wise mixture of Gaussians (GMM). In another work, they demonstrated the efficacy of local block-wise GMM-based data sets for detecting flow instability for rotating stall analysis [14]. Almost all these local distribution-based works model data sets, use regular blocks and summarize each block using the distribution. Here, we propose a novel and improved distribution driven data summarization and demonstrate its efficacy over the existing statistical data summarization methods.

***In situ* processing, analysis, and visualization.** The necessity of *in situ* analysis is becoming more prominent as the size of data output is out-pacing post-processing and visualization capabilities. A comprehensive survey of *in situ* visualization techniques can be found in [4]. Direct visualization of simulation data by performing *in situ* visualization has been used previously. Run-time visualization with LibSim using VisIt was introduced by Whitelock et al. [42]. Lofstead et al. proposed ADIOS [27], and Fabian et al. introduced CATALYST for ParaView [16]. Vishwanath et al. enriched *in situ* analysis by proposing GLEAN [37]. Yu et al. conducted *in situ* visualization of combustion data [47]. A zero copy data structure [44], and an *in situ* eddy census in ocean simulation models [45] were proposed by Woodring et al. However, visualization tasks which require exploratory data analysis can not be done using pure *in situ* approaches. Hence, recently, a new paradigm in *in situ* analysis has gained popularity, where the large-scale data is summarized *in situ* and post-hoc analysis is performed using the summary data. Visualization community has begun to embrace this new technique [10,25]. A sampling-based method for visualization of Cosmology data was used by Woodring et al. [43]. Ahrens et al. adopted an *in situ* image-based approach [2] for feature exploration during post-hoc analysis. Dutta et al. recently enabled efficient *in situ* incremental GMM estimation [14]. In this work, we propose a technique for local distribution-based data summarization and further show the *in situ* applicability of the proposed method using a large-scale CFD simulation.

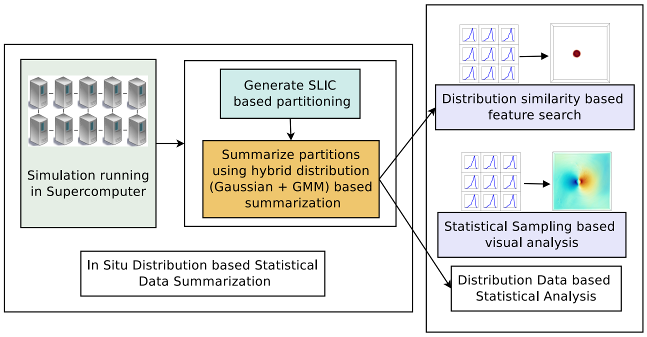

Simulation running in Supercomputer
Generate SLIC based partitioning
Summarize partitions using hybrid distribution (Gaussian + GMM) based summarization
In Situ Distribution based Statistical Data Summarization
Distribution similarity based feature search
Statistical Sampling based visual analysis
Distribution Data based Statistical Analysis

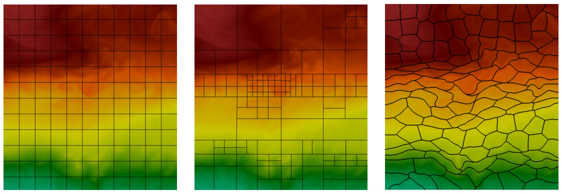
Figure 1: A schematic diagram of our proposed method.

3 Overview

Our main goal is to devise a distribution-based statistical data summarization scheme and show its effectiveness by contrasting it with existing statistical summarization methods. In Figure 1, we present a schematic diagram of the proposed technique. For generating the spatial partitions, we use a fast clustering algorithm SLIC. Then the distribution guided summarization of the partitions are obtained by representing each partition using a single Gaussian distribution or a GMM. The resulting hybrid distribution-based data is then used for analysis and visualization applications. As can be seen from Figure 1, we acknowledge that the ideal time for running the proposed summarization is *in situ*, i.e., during the simulation run. We use several test data sets to demonstrate the efficacy and visual accuracy of our proposed method both quantitatively and qualitatively. Finally, we conduct an *in situ* application study using a large-scale flow simulation.

4 Data Partitioning Schemes

Local region-based data models have recently gained popularity in the visualization community for enabling timely analysis of large-scale scientific data sets. Instead of analyzing at individual point level, the approach uses data blocks as analysis units resulting in computation cost reduction without losing too much information, particularly when the size of the data is very big [38]. Oftentimes, scientific visualization tasks involve exploration of specific phenomena, defined as features, which are found to be spatially connected regions in the data set. In this case, using local region-based analysis allows efficient identification and isolation of features. Furthermore, in some domain specific applications, scientists specifically desire to look at the characteristics of local regions for temporal event discovery, instead of individual points. This is because



(a) Regular block-wise partitioning. **(b)** K-d tree based decomposition. **(c)** SLIC-based partitioning.

Figure 2: Different types of data partitioning schemes.

the behavior of individual points can not display the phenomenon robustly, and may capture false positives during automatic event detection [14]. Hence, local region-based data analysis has its own benefits, but, the accuracy of these methods depend on the quality of partitioning techniques used for generating the domain decomposition. In the following, we present an in depth discussion about the different data partitioning techniques studied in this work.

4.1 Regular Block-wise Partitioning

Regular block-wise partitioning decomposes the domain into equal sized blocks of predetermined dimensions. Due to the simplicity and computational efficacy, this scheme has been widely adopted in many of the previous works involving local region-based data analysis. A regular partitioning of a 2D data is shown in Figure 2a as an illustrative example. Observe that, this scheme does not consider data properties while dividing the domain into smaller subregions. As a result, some partitions will contain data points with a high data value variation. Consequently, statistical summaries estimated from the data points of such blocks will have a higher value spread. Analysis using these summaries will lead to higher uncertainty, since the samples drawn from those block distributions will contain large errors.

4.2 K-d Tree Partitioning

To obtain a more homogeneous decomposition of data, partitioning using k-d trees can be performed. As was discussed in [31], a recursive partitioning of the domain can be obtained by following a top-down subdivision scheme with appropriate termination criterion. Here, we use the information theoretic measure entropy [11] to measure the randomness of a partition. In information theory, the value of Shannon entropy is regarded as a measure of the information content of a probability distribution of a random variable. Formally, information entropy $H(X)$ is measured as:

$$H(X) = -\sum_{i=1}^{n} prob(x_i) log(prob(x_i)) \tag{1}$$

where n is the total number of data points in the partition, $prob(x_i)$ is the probability of data value x_i. It is observed that Shannon entropy increases when the spread of a distribution is higher, i.e., the distribution contains a wide range of data values. In the k-d partitioning scheme, the entropy of the data values of each partition is checked against a predefined threshold value. If the entropy of the partition is higher than the threshold, then the partition is divided into smaller sub-regions to refine the region and reduce the variation. As a result, it guarantees that all the partitions satisfy the predefined homogeneity criterion. An example of this scheme is demonstrated in Figure 2b using a 2D data set. It can be seen that the regions with higher variation have been refined more.

4.3 SLIC-based Partitioning

In this work, we employ a clustering-based data partitioning scheme using a variant of k-means algorithm, called SLIC (Simple Linear Iterative Clustering) [1]. SLIC originally was designed for the generation of superpixels in images, and was also used successfully for the generation of supervoxels in 3D data sets [1, 46]. The fast execution time and state-of-the-art clustering quality make SLIC a suitable choice in our work. Each cluster/supervoxel generated by SLIC is treated as a partition in this work.

Motivation for using SLIC. Compared to traditional k-means clustering, SLIC adopts a local neighborhood-based approach, where similar data points within a local neighborhood are grouped into one cluster. During the optimization stage, from each cluster center, distances only to the points in the predefined neighborhood are compared. This reduces the total number of distance computations significantly by limiting search in a local window. As a result, the algorithm performance is boosted significantly. Furthermore, SLIC uses a weighted distance measure that provides contributions from both the spatial locality of the data points and their scalar value similarities. Due to these properties, SLIC partitions the data domain into smaller sub-regions where each partition contains points which are: (a) spatially as contiguous as possible; and (b) homogeneous in value domain. In Figure 2c, we show an illustrative example of SLIC algorithm applied on a 2D image. As shown, SLIC partitions similar valued data points along non-axis aligned boundaries compared to the methods shown in Figure 2a and 2b. We later demonstrate that summarization using distributions of SLIC partitions achieves superior quality than the previously described partitioning schemes. Below, we briefly discuss the SLIC algorithm.

SLIC algorithm. SLIC requires the user to only provide the expected approximate size of the spatial clusters/partitions. Assuming that the user has provided the spatial size of the partitions as $p \times q \times r$, and if the dimension of the data is $X \times Y \times Z$, then the number of partitions K can be estimated as: $K = (X \times Y \times Z)/(p \times q \times r)$. For finding the K initial cluster centers, the entire data domain is divided into $p \times q \times r$ sized blocks, and the center of each block is selected as its initial cluster center. In the cluster assignment step, each voxel is associated with the nearest cluster center whose search region overlaps with the voxel's spatial location. Since the expected size of a cluster is $p \times q \times r$, the search for similar voxels is done within a volume $2p \times 2q \times 2r$ around each cluster center. This local region-based search during the clustering reduces the total number of distance computations significantly compared to the traditional k-means algorithm, resulting in a overall speed up. Similar to the K-means algorithm, SLIC is an iterative clustering algorithm. During each iteration: (a) Each voxel is associated to its nearest most similar cluster; (b) The cluster centers are recalculated with the updated cluster assignments. For each iteration of SLIC, the difference δ, between the current cluster centers and the previous cluster centers are computed using the L_2 norm between all the cluster centers. If the value of δ is higher than a predefined threshold value, the algorithm moves to its next iteration, otherwise, when δ becomes lower than the threshold, the algorithm terminates.

It is to be noted that, by restricting the search window into a local region for every cluster center, the time complexity of SLIC is significantly reduced compared to a traditional k-means algorithm. The complexity of a k-means algorithm scales with $O(kN)$, whereas, SLIC scales with $O(N)$ [1] for each iteration of the algorithm. This improved computation complexity makes SLIC applicable to large data sets [46] and also attractive for *in situ* environments, where performance is an important factor.

Distance measure. The distance measure used in this algorithm is similar to as was used in [46], and is defined as:

$$dist(i, j) = \alpha \cdot ||C_i - P_j||_2 + (1-\alpha) \cdot |val_i - val_j| \tag{2}$$

Here, C_i is the location of the cluster center i and P_j is the location of point j. val_i and val_j are the scalar values at i^{th} cluster center and j^{th} data point respectively. The mixing weight α is configured based on the importance of spatial vs value components, such that $0 <= \alpha <= 1$, and $\alpha + (1-\alpha) = 1$. Smaller values of alpha will produce higher weightage on the difference of data values than their spatial locations. In Equation 2, as data values and spatial locations can be scaled inconsistently, we normalize the data and normalize spatial distances using the block length to achieve a consistent distance measure.

5 Distribution-Driven Data Modeling and Summarization

As our goal is to achieve a compact and storage efficient statistical summarization of data, we use parametric distribution models for modeling the data in the partitions. Distributions in the form of histograms and Kernel Density Estimators (KDE) require higher storage as compared to parametric distributions like Gaussian mixture model (GMM). The use of Gaussian mixtures as an efficient statistical data summarization has been demonstrated in [14, 15, 26, 39].

For many partitions created in the previous step, a single Gaussian may be a sufficiently accurate representation. Hence, to reduce the storage cost of the distribution-based data summary, we advocate for a hybrid distribution-based data representation scheme. We perform a statistical normality test, D'Agostino's K-squared test [13], on each of the partitions. This test provides a goodness-of-fit measure of departure from normality given the set of data points in a partition. The method uses both kurtosis and skewness to detect the deviation from normality. If a partition satisfies the normality criteria, only a single Gaussian is used to summarize it, otherwise a GMM is estimated for modeling the data in the partition. By using this hybrid distribution summarization scheme, (i.e., Gaussians and GMMs), we achieve a compact statistical summarization of the data without sacrificing the information content of the data. Therefore, for all of the partitioning schemes discussed above in Section 4, we use hybrid summarization scheme for representing the partitions.

Another advantage is the potential reduction in computation cost in creating the distributions for each partitions. Estimation of parameters of a GMM from the given sample points is done using the Expectation Maximization (EM) [5]. This algorithm computes the parameters of a GMM by maximizing a likelihood function using the sample data points. Let us assume that $\chi = \{x_1, x_2, ...x_n\}$ are the set of i.i.d. samples, and θ is the set of parameters. Therefore, the resulting density for the samples $p(\chi|\theta)$ can be expressed as:

$$p(\chi|\theta) = \Pi_{i=1}^{n} p(x_i|\theta) = L(\theta|\chi) \tag{3}$$

Here, $L(\theta|\chi)$ is called the likelihood function, i.e., the likelihood of parameter set θ given the sample data χ. The EM algorithm maximizes this likelihood function and finds $\theta^{\star}$ where,

$$\theta^{\star} = argmax_{\theta} \;\; L(\theta|\chi) \tag{4}$$

Hence, the EM algorithm for calculation of a GMM is computationally costlier compared to estimating the parameters of a single Gaussian distribution. Therefore, if more partitions satisfy the normality test, then the overall computational cost will be reduced, since fewer partitions will employ the EM algorithm. So, by generating coherent and homogeneous partitions, that have low variance via the SLIC method, we can reduce the computation cost and storage of partitions via our hybrid GMM and Gaussian representations.

Algorithm 1 presents the proposed method of statistical data summarization. As discussed above, we employ SLIC for generating the partitions. Then each partition is tested for normality. If the partition satisfies the test, a single Gaussian is used for representing the partition, otherwise a GMM is computed for summarizing it. The final output is reduced and compact hybrid distribution-based data summary. For our following comparisons, we change the step 4 of the above algorithm with a different (regular or k-d tree) partitioning scheme.

Algorithm 1 SLIC-based Statistical Data Summarization

```
 1: Input: Raw data, user specified initial partition dimensions.
 2: Output: Local distribution-based compact summary data.
 3: Initialize cluster/partition centers uniformly over data domain.
 4: Compute SLIC for partition generation.
 5: for all $p$ in $Partitions$ do
 6:     Perform D'Agostino's K-squared normality test.
 7:     if ($p$ satisfies normality test) then
 8:         summarize $p$ using a single Gaussian distribution.
 9:     else
10:         summarize $p$ using a GMM.
11:     end if
12: end for
```

6 Comparative Study among Different Partitioning Techniques

We provide a comprehensive comparative study among the three partitioning methods and demonstrate the efficacy of our proposed method. We consider both storage cost and quality of statistical summarization while comparing these methods. For comparing the quality of statistical summarization, we use sampling-based data reconstruction and visualization as one of our tasks via stochastic sampling-based methodologies for data analysis [28]. We perform sampling on the distribution-based summary data for creating a statistical realization of the raw data. It follows that, with a better quality of the statistical summarization, it will result in a more accurate realization of data with better quality of samples [23]. We employ Monte Carlo sampling for generating a realization of the raw data, as was used in [19, 26].

To estimate the quality of this sampling-based reconstruction, we use Signal-to-Noise Ratio (SNR) for quality comparison. SNR is defined as the dimensionless ratio of the power of a signal to the power of noise in the signal. Hence, higher values of SNR signify better quality. Formally, SNR is defined as:

$$SNR = \frac{P_{signal}}{P_{noise}} \tag{5}$$

where the power of noise is measured by the variance of the error in the reconstructed data. Higher variance of error will decrease the value of SNR. We use the logarithmic decibel scale for SNR: $SNR_{dB} = 10 \cdot log_{10}(SNR)$.

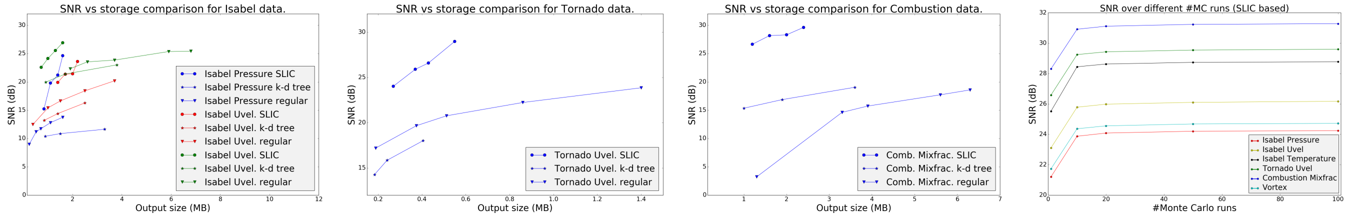
Storage format for different partitioning schemes. For the regular block-wise partitioning scheme, we only store the estimated distribution parameters, i.e., the parameters of the Gaussian distributions (mean and standard deviation) and the parameters of GMMs (means, standard deviations, and weights). Each GMM consists of 3 Gaussian distributions in these experiments. For a partition with a single Gaussian, we keep two floating points for its parameters, and for a partition with a GMM, we use 9 floating points for storing the parameters. Also, we keep a GMM/Gaussian flag for each partition. In case of the k-d tree partitioning scheme, we additionally need to store the ids of two corner point locations of the bounding box for each partition which is stored as integers.

Our SLIC-based method generates irregular partitions and we keep the cluster ids per point as the additional information. Our method is designed for a distributed memory environment and the general assumption is that each node will only process a small subset of data. Furthermore, as we create large homogeneous partitions using SLIC, the range of the cluster ids for each processing node is relatively small. So, we use unsigned shorts for storing the cluster ids which reduces the storage overhead. The point ids for k-d tree partitioning and the cluster ids for the SLIC-based scheme are both stored using zlib compression for further storage reduction.

Storage vs SNR results. The performance of storage vs quality of statistical summarization of: (a) Regular partition based scheme; (b) K-d tree based scheme; and (c) The proposed SLIC-based scheme are presented in Table 1. We have tested several data

Table 1: Experimental results of storage vs SNR (quality) for regular partitioning, k-d tree partitioning, and the proposed SLIC-based partitioning scheme with different parameter configurations. A specific parameter configuration is highlighted in bold from each of the three methods. By observing these three storage vs SNR results, it can be seen that our proposed method achieves superior storage-vs-quality trade-off.

		Regular block partitioning scheme						**K-d tree partitioning scheme**						**SLIC-based partitioning scheme**					
Data set	Raw data size (MB)	**Storage (block size = 3x3x3) (MB)**	**SNR (dB)**	Storage (block size = 4x4x4) (MB)	SNR (dB)	Storage (block size = 5x5x5) (MB)	SNR (dB)	**Storage (entropy th=1.2) (MB)**	**SNR (dB)**	Storage (entropy th=1.5) (MB)	SNR (dB)	Storage (entropy th=1.8) (MB)	SNR (dB)	Storage (Apprx. 5x5x5 points per cluster) (MB)	SNR (dB)	**Storage (Apprx. 6x6x6 points per cluster) (MB)**	**SNR (dB)**	Storage (Apprx. 7x7x7 points per cluster) (MB)	SNR (dB)
Isabel Pressure	12.5	**3.2**	**11.63**	1.5	10.87	0.89	10.39	**1.6**	**13.75**	1.1	12.79	0.71	11.72	1.6	24.63	**1.4**	**21.20**	1.1	19.78
Isabel Uvel	12.5	**2.5**	**16.29**	1.4	14.40	0.85	13.21	**3.7**	**20.18**	2.5	18.42	1.5	16.63	2.2	23.61	**2**	**23.11**	1.7	21.37
Isabel Temperature	12.5	**3.8**	**23.01**	1.7	21.39	0.91	19.96	**6.8**	**25.42**	5.9	25.37	3.7	23.84	1.6	26.92	**1.3**	**25.52**	1	24.14
Tornado Uvel	3.5	**0.41**	**18.01**	0.24	15.84	0.18	14.27	**1.4**	**23.85**	0.86	22.22	0.51	20.76	0.55	28.97	**0.43**	**26.58**	0.37	25.89
Combustion	20.7	**3.6**	**19.05**	1.9	16.87	1.0	15.35	**6.3**	**18.58**	5.6	17.71	3.9	15.73	2.4	29.63	**2**	**28.31**	1.6	28.17
Vortex	8.4	**1.2**	**11.91**	0.90	9.87	0.57	6.37	**7.7**	**19.79**	6.8	17.95	4.7	17.95	1.9	22.68	**1.6**	**21.73**	1.4	20.70



(a) Storage vs SNR comparison of Isabel data.

(b) Storage vs SNR comparison of Tornado data.

(c) Storage vs SNR comparison of Combustion data.

(d) SNR vs #MC runs for SLIC-based summarization scheme.

Figure 3: Figures 3a-3c present storage vs SNR comparison for different data sets. It is observed that using equal or lower storage, proposed SLIC-based method is able to produce better Monte Carlo sampling-based data reconstruction. Figure 3d shows the trend of SNR values with different number of Monte Carlo runs. It can be seen that the SNR values saturate after around 20 Monte Carlo runs. This trend is similar for all the summarization schemes discussed.

sets, described later in our case studies, for conducting these experiments. It is to be noted that, when the partitions are smaller, they are more likely to become more homogeneous compared to bigger partitions. This results in a higher quality of statistical summarization using smaller partitions, but the storage increases. This trend is common for all the 3 methods. The SNR is higher when the size of the partitions are smaller, while the storage is also higher.

The quantitative results of the experiments for all the methods with different parameter configurations are provided in Table 1. As can be seen, we change the block size for regular partitioning scheme to vary the number partitions, and measure the quality of reconstruction in each case by measuring the SNR. In case of the k-d partitioning scheme, we vary the entropy threshold value to obtain a different number of partitions. It is to be noted that, making the entropy threshold higher will result in a decrease of the number of partitions, as well as the storages. However, the SNR will also decrease. Finally, for our proposed scheme, we are able to use bigger partitions (smaller storage) and yet achieve better sampling-based reconstruction quality. The number of clusters are varied by changing the number of points in each cluster. In Table 1, we have highlighted a pair of storage and SNR columns in bold from each of the methods. By comparing these three selected configurations, it can be easily observed that, the proposed method achieves superior storage-to-quality trade-off by producing higher SNR values for all the data sets while using less or comparable storage.

Comparative study among different methods. By using the data presented in Table 1, a line chart based comparison of these three partitioning schemes are presented in Figures 3a-3c. Results of different data sets are shown in separate charts. By studying these 3 charts, a common observation can be made that our proposed SLIC-based technique produces better sampling-based reconstruction of data while using comparable or less storage. It can be seen that by increasing the value of entropy threshold, the storage of k-d tree based scheme can be reduced, however, as mentioned above, it will reduce the SNR, i.e., the reconstruction quality as well. Similarly, we can also create bigger partitions for achieving a better storage in regular block partitioning by sacrificing accuracy. Hence, from Figures 3a-3c, we find that, for similar output storage cost, our proposed method gives superior analysis accuracy among the three methods.

Effect of different numbers of Monte Carlo runs on sampling quality. Since we employ Monte Carlo sampling for generating realization of data from the distribution-based summaries, we further study the effect of different numbers of Monte Carlo runs on the quality of sampling. Ideally, as was shown in [19], more number of Monte Carlo runs would make the reconstruction smoother and the reconstruction quality will increase and eventually will saturate. In Figure 3d, we show that by increasing the number of Monte Carlo runs, and taking an average over all the runs while reconstructing, the reconstruction quality indeed improves. Also, after around 20 Monte Carlo runs, the increase in quality saturates. This trend is observed for all the three summarization schemes.

Comparison of summarization using: (a) Gaussian only; (b) GMM only; and (c) our Hybrid scheme. For the same number of partitions, using only Gaussians for summarization will result in the smallest storage, and using only GMMs will need the highest storage. It is expected that the reconstruction quality will be similar for hybrid scheme vs only GMMs. In Table 2, we show the results of the SLIC-based method (with approximately $6 \times 6 \times 6$ points per cluster) when it is: (a) Gaussian only; (b) GMMs only; and (c) Hybrid distribution-based scheme; using equal number of partitions for each. We find that the sampling-based reconstruction using only GMMs is very similar to the Hybrid scheme. However, when only Gaussians are used, the quality decreases slightly, but it is still superior when compared to the regular partitioning and k-d tree partitioning schemes. This shows that the SLIC partitions are largely homogeneous and a single Gaussian-based summarization can be used per partition when further storage reduction is desired, without compromising the quality much.

Table 2: Comparison of SNR using fixed number of partitions (approx. $6 \times 6 \times 6$ points per cluster) for our SLIC-based scheme when: (a) only Gaussian distributions; (b) only GMM; and (c) Hybrid (Gaussian + GMM) distribution scheme are used for summarization.

Data set	Raw Size (MB)	Gaussian only SNR (dB)	GMM only SNR (dB)	Hybrid (Gaussian + GMM) SNR (dB)
Isabel Pres.	12.5	21.15	21.35	21.20
Isabel Uvel.	12.5	22.98	23.09	23.11
Isabel Temp.	12.5	25.51	25.86	25.52
Torn. Uvel	3.5	26.16	26.85	26.58
Combustion	20.7	28.01	28.42	28.32
Vortex	8.4	21.22	21.72	21.73

7 Visual Analysis

Distribution-based summary data can be used in two ways for analyzing scientific data sets. The first technique is by directly exploiting the local statistical properties of the data for distribution-driven classification and feature search. This method does not require any sampling and analyzes the distributions of the local regions directly for classification. The second method is using Monte Carlo sampling-based reconstruction of a statistical realization of the data for visual analytics. For all the visualizations ParaView [3] was used for rendering the results, and we used $3 \times 3 \times 3$ block-size for regular-partitioning scheme, entropy threshold of 1.2 for k-d partitioning, and approximate partition size of $6 \times 6 \times 6$ points per partition for our slic-based method. In the experiments, we show that, with lower or comparable storage cost, as reported previously in Table 1, our method produces better and more accurate visual quality.

7.1 Distribution-Driven Stochastic Feature Search

Scientific data sets contain features that are not well defined in the value domain and it is difficult to define such features using precise threshold values. Several previous works [14, 15, 29, 40] have shown the use of distributions to represent such features probabilistically. In the absence of a precise value range-based feature descriptor, stochastically-defined features can be searched by using local distribution-based data summaries. Local regions (partitions) containing similar distributions that of the target distribution will be detected. Here we show that by using our distribution-based summarization, a more accurate and refined feature searching can be performed. For measuring the similarity between distributions, we use the Earth Mover's Distance (EMD), defined by the minimal transport effort to match two distribution shapes. To compute the EMD for 1D distributions, we use the *match distance* as the ground distance [35], since, the EMD then can be estimated by the absolute difference between the cumulative distribution functions (CDF) of the distributions [41].

7.1.1 Feature Search in Tornado Data Set

Our first experiment studies feature searching in a Tornado data set with spatial dimensions of $96 \times 96 \times 96$, and velocity vectors at each grid point, generated by an analytical equation [12]. The data set has 50 time steps simulating a tornado-like vortex structure. For this case study, we use the U-velocity field.

As seen from Figure 4a, the primarily high values of U-velocity provides a structure of the tornado, which is the feature of interest. For highlighting the region of interest, the users select a small 3D box on this region, as shown in Figure 4a to easily select their region of interest [15]. We collect the data points in this box for defining the target feature distribution as a GMM. The estimated feature GMM is shown on the right of Figure 4a. Using an user specified fixed threshold of 0.1 on the normalized EMD-based distance field, the detected region is extracted and visualized from the statistical summary data. Figure 4b and 4c show the results obtained from regular partitioning scheme (block size = $3 \times 3 \times 3$), and k-d tree partitioning scheme (entropy threshold = 1.2). The identified regions contain blocky artifacts on the boundaries as observed from the results. In contrast, our SLIC-based method partitions data by its local homogeneity which preserves the feature boundaries more accurately.

7.1.2 Feature Search in Vortex Data Set

Our second case study shows the result of distribution-driven feature search in a Vortex data set, which is a pseudo-spectral simulation of coherence vortex structures. The spatial dimensions of this data set is $128 \times 128 \times 128$. The scalar field used in the data is vorticity magnitude containing several tubular vortex cores, which are the features of interest.

The high vorticity values roughly correspond to the vortex features, which can be seen in the Figure 5a. By following a similar technique as discussed in Section 7.1.1, the target feature GMM is obtained and shown on the right of Figure 5a. Compared to the previous Tornado case study, identifying features in this data set is not easy since there are many features which show similar data properties, and such features are located separately across the spatial domain as seen in Figure 5a. Furthermore, there are several vortex features , as shown by black dotted lines in Figure 5a, that are very small and hence, challenging to be detected. From the results presented in Figures 5b, 5c, and 5d, we see that our SLIC-based partitioning method is the best in extracting those vortex features among the three techniques. The EMD threshold of 0.23 was used for the extraction of the matched regions. Also, from Figures 5b and 5c, it is observed that both the regular block-wise scheme and k-d tree based scheme detect the small features less accurately compared to our proposed method. The shape of those small features gets distorted as highlighted by red dotted lines in Figures 5b and 5c, whereas the proposed method is able to identify these fine features with high accuracy.

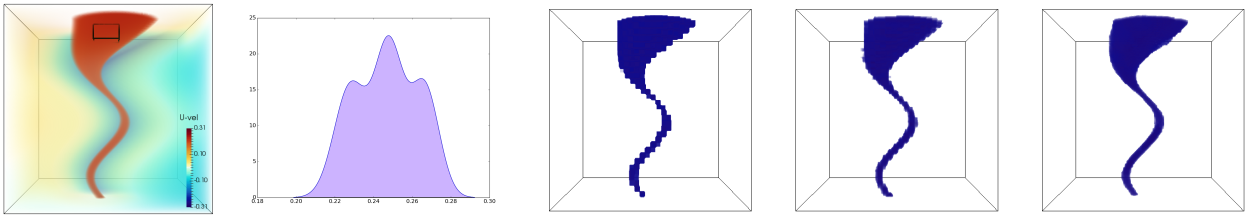
7.1.3 Feature Search in Hurricane Isabel Data Set

Hurricane Isabel data is a multivariate time-varying data consisting of 13 scalar fields. The data set is a courtesy of NCAR and the U.S. National Science Foundation (NSF), and was created using the Weather Research and Forecast (WRF) model. The resolution of the grid for each time step is $250 \times 250 \times 50$ and there are total 48 time steps. In this study, we use the Pressure field of the data set.

We use the low pressure region which is known as the eye of the hurricane and is an important feature in the data. We show the selected region using a small 3D box and the estimated target distribution in Figure 6a. The results of the detected feature using different schemes are depicted in Figures 6b, 6c, and 6d. The EMD threshold of 0.25 was used for this experiment. From the zoomed view of the detected regions, we see that the blocky artifacts due to axis-aligned partitioning is visible in both Figures 6b and 6c on the boundaries. However, we obtain a much smoother and refined feature matching using our proposed summarization technique as observed from Figure 6d.

7.2 Sampling-based Data Visualization

Monte Carlo sampling-based reconstruction for visualizing distribution-based data sets was used previously in [19, 26]. In this section, we demonstrate that by summarizing the data using our proposed scheme, a more accurate sampling-based visualization can be achieved compared to the other discussed methods. We use zlib compression for storing the point ids in k-d partitioning, and cluster ids in SLIC-based partitioning. Decompression is done in memory during runtime when cluster information for reconstruction is required for visualization.



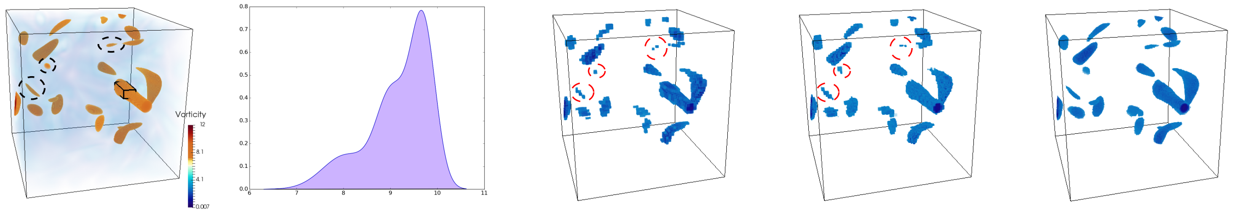
(a) Feature selection in Tornado data set. The estimated target feature distribution is shown on the right. The feature is modeled using a mixture of Gaussians.

(b) Distribution similarity-based identified feature using regular block partitioning.

(c) Distribution similarity-based identified feature using k-d tree partitioning.

(d) Distribution similarity-based identified feature using our SLIC-based partitioning.

Figure 4: Distribution data driven probabilistic feature search in Tornado data set.



(a) Feature selection in Vortex data set. The estimated target feature distribution is shown on the right. The feature is modeled using a mixture of Gaussians.

(b) Distribution similarity-based identified feature using regular block partitioning.

(c) Distribution similarity-based identified feature using k-d tree partitioning.

(d) Distribution similarity-based identified feature using our SLIC-based partitioning.

Figure 5: Distribution data driven probabilistic feature search in Vortex data set.

7.2.1 Visual Analysis using Hurricane Isabel Data Set

This case study uses the U-velocity field of Hurricane Isabel data set, which was used earlier in Section 7.1.3. Figure 7 shows the volume rendered images from the reconstructed data using different methods. In Figure 7d we present the result of raw data and a zoomed view of the core region of the storm showing the high and low wind speed. This is an important region for U-velocity, since the wind velocity can reflect the power of the storm. As seen from the zoomed view on the right of Figure 7a (regular block scheme), the image produces checker-box-like artifacts (as shown by black circle in Figure 7a). Note that, this image is produced using a block size of $3 \times 3 \times 3$. If we increase the block size, these artifacts will become even more prominent which further reduces the visual quality. In comparison, the k-d tree based reconstructed data generates a comparatively smoother result with fewer artifacts (highlighted with black dotted lines in Figure 7b), however, a low entropy threshold of 1.2 was used to achieve it which led to increased storage (see column 9 of Table 1). Finally, Figure 7c depicts the result of our proposed SLIC-based partitioning scheme (uses approximately $6 \times 6 \times 6$ points per cluster), which produces the closest visual quality to the raw data.

7.2.2 Visual Analysis using Turbulent Combustion Data Set

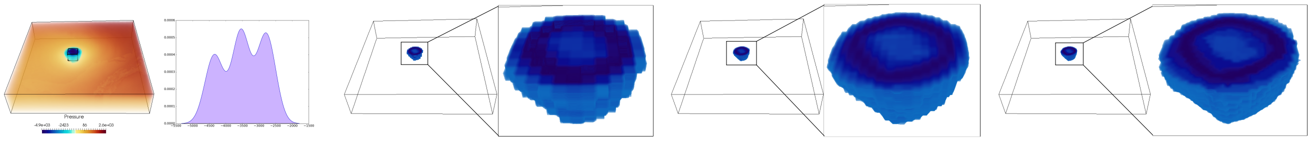
The Combustion data set is a time-varying turbulent simulation data set containing 5 chemical variables. The spatial resolution of each variable is $240 \times 360 \times 60$. The data set was made available by Dr. Jacqueline Chen at Sandia Laboratories through US Department of Energy's SciDAC Institute for Ultra-scale Visualization. We used the mixture fraction variable, which represents the proportion of oxidizer mass and fuel in the data.

Visualizations generated by different summarization methods using the mixture fraction variable of Combustion data set are shown in Figure 8. From the reconstructed image using regular partitioning scheme using block size $3 \times 3 \times 3$, we see checker-box-like discontinuities in Figure 8a marked with red dotted lines. The k-d tree based reconstruction technique is able to reduce this checker-box-artifact. However, the result still shows some differences on the boundary of the flame structures (highlighted with red dotted lines in Figure 8b) when compared to the raw data in Figure 8d. In particular, the k-d tree algorithm produces a partition (red dotted region in top right in Figure 8b) which only covers a small portion of the flame structure. The rest of the region is considered background, containing homogeneous values. As a result, this region was not partitioned further by the k-d tree since it satisfied the entropy-based termination criterion. During reconstruction, this causes higher error making it impossible to recover the correct boundary of the flame structure. Note that, a smaller entropy threshold will divide this region into smaller partitions, thus reducing this artifact, but consequently the storage cost will increase.

The visualization produced by the proposed SLIC-based partitioning scheme, depicted in Figure 8c, was produced using approximately $6 \times 6 \times 6$ points per cluster. With the smallest storage, the SLIC-based visualization matches the raw data the best, preserving the overall flame structures on both the sides.

8 *In Situ* Application Study, Performance, and Expert Feedback

In this section, we present a domain study using a large-scale computational fluid dynamics (CFD) simulation code, TURBO [8, 9], and demonstrate the applicability of our method for *in situ* environments. TURBO is a high-resolution, Navier-Stokes based, time accurate CFD code, developed at NASA, and is used to study the flow instability in transonic jet engine compressors. Our domain expert studies the characteristics of Pressure values for detecting the inception of flow instability. Figures 9a-9d show the sampling-based reconstructed results of Pressure variable, where our proposed method produces higher quality reconstruction compared to the other methods. The image generated using regular partitioning and k-d partitioning contain artifacts highlighted with black dotted lines in Figure 9a and 9b. Furthermore, we study the storage vs SNR of the three methods and present the results in Figure 9e. We



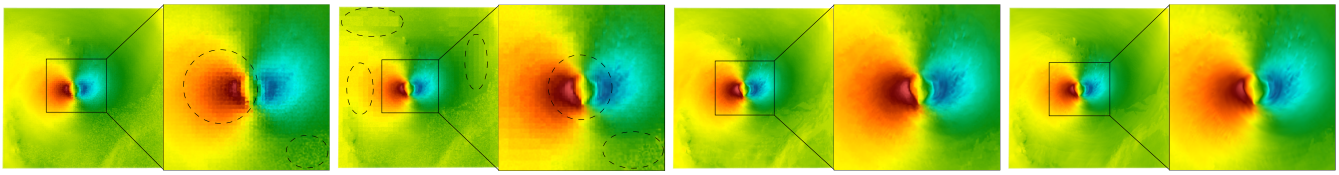
(a) Feature selection in Isabel data set. The estimated target feature distribution is shown on the right. The feature is modeled using a mixture of Gaussians.

(b) Distribution similarity-based identified feature using regular block partitioning.

(c) Distribution similarity-based identified feature using k-d tree partitioning.

(d) Distribution similarity-based identified feature using our SLIC-based partitioning.

Figure 6: Distribution data driven probabilistic feature search in Hurricane Isabel data set.



(a) Reconstruction using regular block partitioning scheme. The block size used is $3 \times 3 \times 3$. A zoomed view is shown on the right.

(b) Reconstruction using k-d tree based scheme. Entropy threshold of 1.2 is used for this experiment. A zoomed view is shown on the right.

(c) Reconstruction using proposed SLIC-based scheme. Relatively large cluster size (approx. $6 \times 6 \times 6$ points per cluster) is used. A zoomed view is shown on the right.

(d) Raw data (Ground truth). A zoomed view is shown on the right for better visual comparison.

Figure 7: Visual comparison of U-velocity of Hurricane Isabel data. The reconstructed fields are generated using Monte Carlo sampling of distribution-based summarized data.

Table 3: *In situ* timings of our proposed method.

Simulation (hrs)	Simulation raw I/O (hrs)	*In situ* analysis (hrs)	*In situ* I/O (hrs)
13.217	2.06	1.822	0.015

observe that with equal storage, our method achieves much higher quality than the other methods. However, the size of raw data output of a single simulation is quite large which makes the analysis cumbersome and overwhelming for the expert.

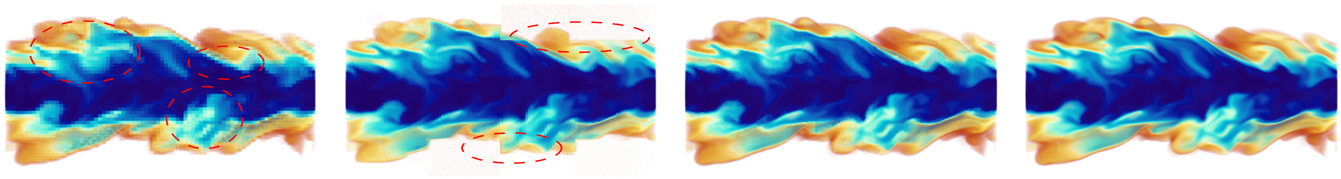
We applied our method for summarizing Pressure variable *in situ* which only stored the distribution-based summary data. Our *in situ* study was done using a cluster, Oakley [6], at the Ohio Supercomputer Center, which contains 694 nodes with Intel Xeon x5650 CPUs (12 cores per node), and 48 GB of memory per node. The simulation was run with 328 cores for the study, and we ran it for 2 revolutions, resulting in 7200 time steps. *In situ* call was made at every 10^{th} time step which required us to process 1.008 TBs of data for 720 time steps. Note that, the expert used to only write out raw data at 25-30 time steps without the *in situ* capability. We summarized the data of the rotor section of the model, by directly accessing the simulation memory without any additional data copy. During *in situ* processing, we generated the partitioning using slic and summarized the partitions with our hybrid distribution-based scheme. The raw simulation outputs 5 variables in multi-block plot3d format, and the raw data size for the rotor section is 690 MB per time step, i.e., 496.8 GB for just 2 revolutions. The domain of the compressor consists of 36 blocks (blade passages), and the spatial resolution of each block is $151 \times 71 \times 56$. In contrast, the size of our summarized data for Pressure variable is only 10.8 GB, i.e., around 54 GB for all 5 variables, using SLIC-based summarization with approximately $6 \times 6 \times 6$ points per partition, resulting in a significantly smaller data. Table 3 shows the timings of the *in situ* run for this study. We see that our method takes about 13.5% of the simulation time for analyzing the data and summarizing it. In contrast, post-hoc SLIC-based partitioning and summarization on a standard Linux machine with an Intel core i7-2600 CPU, 16 GB of RAM, and 1 TB HDD using OpenMP parallelization, takes 73.5 secs on average per time step, i.e., about 14.7 hrs for processing 720 time steps. Furthermore, the computation time for Monte Carlo sampling for all the 720 time steps took 2.96 hrs including the I/O time.

Domain expert feedback. We presented the results to the domain scientist. The expert agreed that, the reduced summary data accelerates the post-hoc analysis, which can be used as a replacement of the raw data for exploration. The expert was particularly impressed by the reconstruction quality that we achieved with our method over the existing techniques. Also, the expert acknowledged that, by transforming the data into a distribution-based representation, a wide range of visual-analytics can be done using it. With our *in situ* data triage and summarization, the expert now can keep higher temporal resolution of data by storing more time steps than before, which will help in a more precise temporal event detection. Hence, the expert feels that the additional computational time spent for *in situ* analysis is well justified, given the benefits it offers during exploratory post-hoc analysis. Finally, the domain expert also suggested to extend our method for multivariate and vector fields which will increase the usefulness of the method.

9 Discussion

The regular partitioning requires minimum storage for same number of partitions among these techniques, because the partition bounds are implicit, so, no additional storage is necessary. However, the quality of sampling-based data reconstruction, and probabilistic feature analysis using regular block-based summary data is found to be less effective, since this method does not consider data value coherency, resulting in partitions with high value variance. The key difference between regular partitioning, k-d tree partitioning, and SLIC-based scheme is that, the last two methods aim at reducing value variation while creating the spatial partitions. Nevertheless, for finding spatially homogeneous regions, k-d partitioning often divides the data into many smaller axis-aligned regions which causes higher storage. Note that, we can reduce the storage of k-d based scheme by changing the k-d tree decomposition termination criterion, but, that will also reduce the summarization quality.

Our SLIC-based partitioning works by generating irregular partition shapes. With this, SLIC captures data homogeneity better than the other two methods. It minimizes data value variation inside each partition, and enables more accurate statistical data summarization. From the three selected parameter configurations highlighted in Ta-



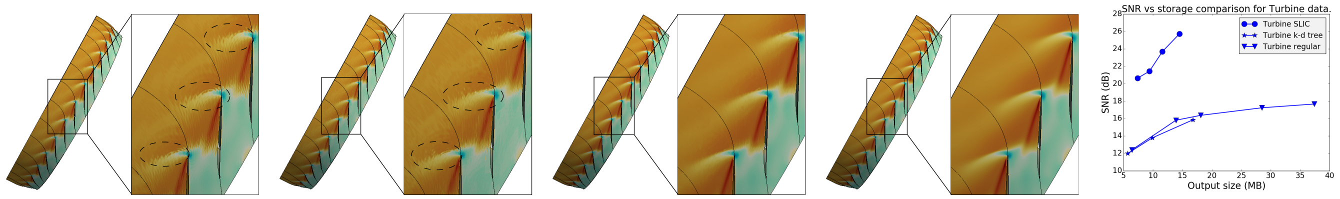
(a) Reconstruction using regular block partitioning scheme. The block size used is $3 \times 3 \times 3$.

(b) Reconstruction using k-d tree based scheme. Entropy threshold of 1.2 is used for this experiment.

(c) Reconstruction using proposed SLIC-based scheme. Relatively large cluster size (approx. $6 \times 6 \times 6$ points per cluster) is used.

(d) Raw data (Ground truth).

Figure 8: Visual comparison of Mixture Fraction of Combustion data. The reconstructed fields are generated using Monte Carlo sampling of distribution-based summarized data.



(a) Reconstruction using regular partitioning scheme. The block size is $3 \times 3 \times 3$. A zoomed view is shown on the right.

(b) Reconstruction using k-d tree based scheme. Entropy threshold of 1.2 is used. A zoomed view is shown on the right.

(c) Reconstruction using SLIC-based scheme. Approx. $6 \times 6 \times 6$ points per cluster is used. A zoomed view is shown on the right.

(d) Raw data (Ground truth). A zoomed view is shown on the right for better visual comparison.

(e) Storage vs SNR comparison of Turbine data.

Figure 9: Figures 9a-9d: visual comparison of Pressure field of Turbine data set. The reconstructed fields are generated using Monte Carlo sampling of summarized data. Figure 9e: storage vs quality comparison of turbine data set. It is observed that, with similar storage, proposed method produces more accurate visual quality.

ble 1, we find that, SLIC-based summarization preserves the statistical data properties more accurately, reflected by the best SNR-to-storage ratio. Using larger partitions, we effectively summarize a smaller number of partitions when compared to the other two methods, which is the primary reason that we have the best SNR-to-storage ratio. We achieve superior partitioning through irregular shaped cluster representation and compact distribution-based summarization with compressed cluster id information.

However, the introduction of irregular shaped partitions in our method has resulted in an storage overhead of cluster ids per point, which can be regarded as a potential limitation. An improved method for storing cluster information will make our method even more storage efficient. Finally, we have successfully applied our method to a large-scale CFD data set to demonstrate the *in situ* capability of our method. Positive feedback from our domain expert further show the effectiveness of our method for summarizing large-scale data sets for flexible post-hoc analysis.

10 Conclusions and Future Work

We present a local homogeneity-driven partitioning based stochastic data summarization technique for large-scale data analysis and visualization. We demonstrate the efficacy of our method by contrasting our proposed method with the existing statistical data summarization techniques. We show that our method works *in situ* and preserves statistical data properties through a superior partition-based summarization, which allows effective probabilistic feature analysis and visualization. In this work, we process each time step separately. However, we wish to extend our method for partitioning multiple time steps together such that a time-window will share the same partitioning. This scheme will further reduce the storage as multiple time steps will share same clustering information. Also, we want to extend this work for summarizing of multivariate data sets, and employ it on other scientific applications for assisting domain scientists.

Acknowledgments

This work was supported in part by NSF grants IIS- 1250752, IIS-1065025, and US Department of Energy grants DE- SC0007444, DE-DC0012495, program manager Lucy Nowell.

References

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

[2] J. Ahrens, S. Jourdain, P. O'Leary, J. Patchett, D. H. Rogers, and M. Petersen. An image-based approach to extreme scale in situ visualization and analysis. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '14, pp. 424–434. IEEE Press, Piscataway, NJ, USA, 2014. doi: 10.1109/SC.2014.40

[3] U. Ayachit. *The ParaView Guide: A Parallel Visualization Application*. Kitware Inc., 4.3 ed., 2015. ISBN 978-1-930934-30-6.

[4] A. C. Bauer, H. Abbasi, J. Ahrens, H. Childs, B. Geveci, S. Klasky, K. Moreland, P. O'Leary, V. Vishwanath, B. Whitlock, and E. W. Bethel. In situ methods, infrastructures, and applications on high performance computing platforms. *Computer Graphics Forum*, 35(3):577–597, 2016. doi: 10.1111/cgf.12930

[5] J. Bilmes. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical report, 1998.

[6] O. S. Center. Oakley supercomputer. `http://osc.edu/ark:/19495/hpc0cvqn`, 2012.

[7] A. Chaudhuri, T. H. Wei, T. Y. Lee, H. W. Shen, and T. Peterka. Efficient range distribution query for visualizing scientific data. In *2014 IEEE Pacific Visualization Symposium*, pp. 201–208, March 2014. doi: 10.1109/PacificVis.2014.60

[8] J. Chen, R. Webster, M. Hathaway, G. Herrick, and G. Skoch. Numerical simulation of stall and stall control in axial and radial compressors. In *44th AIAA Aerospace Sciences Meeting and Exhibit*. American In-

stitute of Aeronautics and Astronautics, 2006. doi: 10.2514/6.2006-418

[9] J.-P. Chen, M. D. Hathaway, and G. P. Herrick. Prestall behavior of a transonic axial compressor stage via time-accurate numerical simulation. *Journal of Turbomachinery*, 130(4):041014, 2008. doi: 10.1115/1.2812968

[10] H. Childs. Data exploration at the exascale. *Supercomputing frontiers and innovations*, 2(3), 2015.

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

[12] R. Crawfis and N. Max. Texture splats for 3d scalar and vector field visualization. In *Visualization, 1993. Visualization '93, Proceedings., IEEE Conference on*, pp. 261–266, Oct 1993. doi: 10.1109/VISUAL.1993.398877

[13] R. B. D'agostino, A. Belanger, and R. B. D. Jr. A suggestion for using powerful and informative tests of normality. *The American Statistician*, 44(4):316–321, 1990. doi: 10.1080/00031305.1990.10475751

[14] S. Dutta, C. M. Chen, G. Heinlein, H. W. Shen, and J. P. Chen. In situ distribution guided analysis and visualization of transonic jet engine simulations. *IEEE Transactions on Visualization and Computer Graphics*, PP(99):1–1, 2016. doi: 10.1109/TVCG.2016.2598604

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

[16] 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)*, pp. 89–96, 2011. doi: 10.1109/LDAV.2011.6092322

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

[18] Y. Gu and C. Wang. TransGraph: hierarchical exploration of transition relationships in time-varying volumetric data. *IEEE Trans. on Vis. and Comp. Graphics*, 17(12):2015–24, 2011. doi: 10.1109/TVCG.2011.246

[19] H. Guo, W. He, T. Peterka, H. W. Shen, S. M. Collis, and J. J. Helmus. Finite-time lyapunov exponents and lagrangian coherent structures in uncertain unsteady flows. *IEEE Transactions on Visualization and Computer Graphics*, 22(6):1672–1682, June 2016. doi: 10.1109/TVCG.2016.2534560

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

[21] D. Kao, A. Luo, J. L. Dungan, and A. Pang. Visualizing spatially varying distribution data. In *Proceedings of the Sixth International Conference on Information Visualisation, 2002*, pp. 219–225, 2002.

[22] J. M. Kniss, R. V. Uitert, A. Stephens, G. S. Li, T. Tasdizen, and C. Hansen. Statistically quantitative volume visualization. In *VIS 05. IEEE Visualization, 2005.*, pp. 287–294, Oct 2005. doi: 10.1109/VISUAL.2005.1532807

[23] U. Kohler and F. Kreuter. *Data Analysis using Stata, 2nd Edition*. StataCorp LP, 2009.

[24] T.-Y. Lee and H.-W. Shen. Efficient local statistical analysis via integral histograms with discrete wavelet transform. *IEEE Trans. on Vis. and Comp. Graphics*, 19(12):2693–702, 2013. doi: 10.1109/TVCG.2013.152

[25] 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*, pp. 51–58, 2014. doi: 10.1109/LDAV.2014.7013204

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

[27] J. F. Lofstead, S. Klasky, K. Schwan, N. Podhorszki, and C. Jin. Flexible IO and Integration for Scientific Codes Through the Adaptable IO System (ADIOS). In *Proceedings of the 6th International Workshop on Challenges of Large Applications in Distributed Environments*, CLADE '08, pp. 15–24. ACM, 2008. doi: 10.1145/1383529.1383533

[28] S. Lohr. *Sampling: Design and Analysis*. Advanced (Cengage Learning). Cengage Learning, 2009.

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

[30] A. Luo, D. Kao, and A. Pang. Visualizing spatial distribution data sets. In *Proceedings of the Symposium on Data Visualisation 2003*, VISSYM '03, pp. 29–38, 2003.

[31] B. Nouanesengsy, J. Woodring, J. Patchett, K. Myers, and J. Ahrens. Adr visualization: A generalized framework for ranking large-scale scientific data using analysis-driven refinement. In *Large Data Analysis and Visualization (LDAV), 2014 IEEE 4th Symposium on*, pp. 43–50, Nov 2014. doi: 10.1109/LDAV.2014.7013203

[32] K. Pöthkow and H.-C. Hege. Nonparametric models for uncertainty visualization. In *Proceedings of the 15th Eurographics Conference on Visualization*, EuroVis '13, pp. 131–140, 2013. doi: 10.1111/cgf.12100

[33] K. Potter, J. Kniss, R. Riesenfeld, and C. R. Johnson. Visualizing summary statistics and uncertainty. *Computer Graphics Forum (Proceedings of Eurovis 2010)*, 29(3):823–831, 2010.

[34] K. Potter, J. Krüger, and C. Johnson. Towards the visualization of multi-dimensional stochastic distribution data. In *Proceedings of The International Conference on Computer Graphics and Visualization (IADIS) 2008*, 2008.

[35] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000. doi: 10.1023/A:1026543900054

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

[37] V. Vishwanath, M. Hereld, and M. E. Papka. Toward simulation-time data analysis and i/o acceleration on leadership-class systems. In *2011 IEEE Symposium on Large Data Analysis and Visualization (LDAV)*, pp. 9–14, 2011. doi: 10.1109/LDAV.2011.6092178

[38] C. Wang, H. Yu, and K.-L. Ma. Importance-driven time-varying data visualization. *IEEE Trans. on Vis. and Comp. Graphics*, 14(6):1547–1554, 2008.

[39] Y. Wang, W. Chen, J. Zhang, T. Dong, G. Shan, and X. Chi. Efficient volume exploration using the gaussian mixture model. *Visualization and Computer Graphics, IEEE Transactions on*, 17(11):1560–1573, Nov 2011.

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

[41] M. Werman, S. Peleg, and A. Rosenfeld. A distance metric for multidimensional histogram. *CVGIP: Graphical Models and Image Processing*, 32(3):328–336, 1983. doi: 10.1177/014662168300700106

[42] 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*, EGPGV '11, pp. 101–109. Eurographics Association, 2011. doi: 10.2312/EGPGV/EGPGV11/101-109

[43] 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*, pp. 1151–1160. Eurographics Association, 2011. doi: 10.1111/j.1467-8659.2011.01964.x

[44] J. Woodring, J. Ahrens, T. J. Tautges, T. Peterka, V. Vishwanath, and B. Geveci. On-demand unstructured mesh translation for reducing memory pressure during in situ analysis. In *Proceedings of the 8th International Workshop on Ultrascale Visualization*, pp. 3:1–3:8. ACM, 2013. doi: 10.1145/2535571.2535592

[45] J. Woodring, M. Petersen, A. Schmeißer, J. Patchett, J. Ahrens, and H. Hagen. In situ eddy analysis in a high-resolution ocean climate model. *IEEE Trans. on Vis. and Comp. Graphics*, 22(1):857–866, 2016. doi: 10.1109/TVCG.2015.2467411

[46] J. Xie, F. Sauer, and 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*, pp. 17–24, Oct 2015. doi: 10.1109/LDAV.2015.7348067

[47] H. Yu, C. Wang, R. W. Grout, J. H. Chen, and K. L. Ma. In situ visualization for large-scale combustion simulations. *IEEE Computer Graphics and Applications*, 30(3):45–57, 2010. doi: 10.1109/MCG.2010.55