Probabilistic Data-Driven Sampling via Multi-Criteria Importance Analysis

Ayan Biswas, Soumya Dutta, Earl Lawrence, John Patchett, Jon C. Calhoun, and James Ahrens

Abstract—Although supercomputers are becoming increasingly powerful, their components have thus far not scaled proportionately. Compute power is growing enormously and is enabling finely resolved simulations that produce never-before-seen features. However, I/O capabilities lag by orders of magnitude, which means only a fraction of the simulation data can be stored for post hoc analysis. Prespecified plans for saving features and quantities of interest do not work for features that have not been seen before. Data-driven intelligent sampling schemes are needed to detect and save important parts of the simulation while it is running. Here, we propose a novel sampling scheme that reduces the size of the data by orders-of-magnitude while still preserving important regions. The approach we develop selects points with unusual data values and high gradients. We demonstrate that our approach outperforms traditional sampling schemes on a number of tasks.

Index Terms—Importance sampling, data reduction, error quantification, feature preservation.

1 Introduction

The exascale era is almost upon us. From such powerful compute capabilities, our ability to perform science via large-scale simulations is going to increase multi-fold. Simulations are set to produce unprecedented amounts of data by resolving very fine resolutions in space and time. The traditional post hoc data visualization workflow (where the data from the simulations are transferred to permanent storage for detailed analysis) is set to become obsolete because I/O capabilities have not increased at the same rate as computation speed. This necessitates online or *in situ* processing such that the newer findings are not lost due to I/O limitations and storage constraints. Thus, *in situ* analysis of large-scale simulations, where the analysis is performed while the data is being produced by the simulation, has become an important part of data analysis and the visualization pipeline in the past decade.

When exascale machines are operating at peak frequency and producing data from exascale-enabled simulations, limited I/O bandwidth means only a small fraction of the data produced is saved out to a disk for post hoc processing. Knowing what is important and which regions to save when a simulation is running is non-trivial. In the past, data importance has been assigned by user-driven importance techniques [1], [2], [3]. These methods are generally very specific to a given simulation and primarily work well for a limited set of conditions. If simulations are producing new (potentially unseen) features, then a new set of importance criteria (what is a feature or region of interest?) needs to be added to the *in situ* code for accurate detection of such important events. Despite the need, even with recent efforts for *in situ* data analysis, generic data saliency computation methods are mostly lacking.

For data reduction, compared to the sophisticated data modeling approaches [4], [5], [6], sampling can provide a representation of the complete data with much smaller memory and computation requirements. Sampling of large-scale datasets has been primarily performed based on the popular uniform and/or random selection. These methods are generic and heavily employed because of their simplicity, but these methods generally do not assign varying importance to the individual data points when used for spatial sampling. Although the resulting samples mostly preserve statistical properties (such as mean and variance) of the original data, these methods often overlook a key fact for visualization purposes—not all data points are equally important [7], [8], [9], [10], [11], [12].

For scientific datasets, many analysis and visualization tasks are driven by the notion of features. Apart from knowing distributional properties of the data, scientists are often inclined to explore the small regions-of-interest (high temperature, low pressure, etc.). Query-driven visualization methods [13], [14] have been a popular choice for such feature-based exploration tasks. For these applications, dataset feature regions are generally more important than non-feature regions. To facilitate such query-based visualization applications for samples of large-scale datasets, it is necessary to assign more importance to the more likely regions of interest.

In this work, we propose the use of generic data-driven importance-based sampling algorithms that can later be used for fast user queries and feature-based reconstruction. We investigate the existing sampling methods and propose a novel data-driven sampling method that incorporates the knowledge of data importance based on local and global data properties.

In this paper, we adopt a data-driven approach to identify the data values that are probabilistically more salient. Our proposed sampling method converts a structured scientific dataset into a point cloud that can later be used for user

- *A. Biswas is with Los Alamos National Laboratory.
E-mail: see http://ayanbiswas.net/contacts.html*
- *A. Biswas, S. Dutta, E. Lawrence, J. Patchett and J. Ahrens are with Los Alamos National Laboratory.*
- *J. C. Calhoun is with Clemson University.*

Manuscript received April 19, 2005; revised August 26, 2015.

queries and visualization. For generic importance analysis of a scientific dataset, we employ multiple criteria based on data properties that encompass both global distributional aspects as well as local data smoothness. Using this importance modeling, for a given storage constraint, the samples are identified using fast local computation. For effective use of the storage, the locations of the point samples generated via sampling are condensed separately from the field values using a variant of the sparse coding method. We apply our method to various real-world scientific datasets to show the usefulness of our algorithm. Using these importance-driven samples, we demonstrate our ability to achieve fast query-driven visualization and feature-based local data reconstruction. We compare our method with other sampling methods to illustrate its superior qualities.

Our contributions are multi-fold:

- We propose a novel data sampling technique that combines both local and global data properties but is still light-weight.
- Given a storage constraint, we help ensure that features of scientific data are well preserved in the resulting data samples.
- We evaluate and provide a detailed study of our algorithm when applied across various scientific datasets.

2 Related Works

Sampling-Based Data Analysis and Visualization: The visualization community has developed several methods that employ various data sampling techniques to reduce the size of the very large-scale datasets so that visualization and analysis can be performed in a timely manner. Woodring et al. [15] proposed a stratified random sampling-based scheme for the summarization of cosmology simulations, which enabled interactive post hoc visualization. Wei et al. [16] extended traditional stratified random sampling and used bitmap indexing and information theoretic measures for creating *in situ* compressed sub-sampled datasets. In another work, Su et al. [17] utilized bitmap indexing for performing efficient data sampling. Park et al. [18] proposed a visualization-aware sampling technique, which sampled a very small fraction of data for producing an accurate visualization. Since this technique was optimized for the scatter plot-based and map plot-based visualization techniques, it cannot guarantee high-quality samplings for producing general purpose visualizations. Nguyen and Song [19] proposed a centrality clustering-based data sampling scheme for improving simple random sampling. Use of information theoretic measures has also been found useful for data subsampling. Several researchers have used information entropy to select a subset of data through maximizing entropy in order to find a good representative of the datasets [20], [21], [22]. Following the similar principle of information theory, Biswas et al. [7] proposed a scheme of sampling large-scale datasets *in situ* for producing a subset of informative data that preserves the important features. They used the probability of data values to assign importance to data points while sampling from the simulation output. In a more recent work, Rapp et al. [23] proposed a sampling approach for scattered datasets that identifies a representative subset of points preserving blue noise properties. Our proposed method is intended for regular grid datasets and designed to capture the feature regions of the scientific datasets, even at very low sampling rates.

Large-Scale Data Reduction and Visualization: The size of simulation data keeps increasing, therefore, scientists are looking at various techniques for data reduction to make interactive analysis and visualization tractable. Several researchers have suggested direct visualization of the data *in situ*, that is, when the simulation is running and the data is at the supercomputer memory. For direct *in situ* visualization, several general-purpose *in situ* infrastructures have been added into existing visualization frameworks [24], [25], [26], [27], [28].

It is to be noted that a subset of the data analysis and visualization tasks, where user interaction and feedback are necessary, cannot be performed *in situ* because of the time and resource constraints. So, post hoc visualization is still relevant, and efficient data reduction techniques are essential to be able to explore large-scale datasets interactively in the post hoc analysis phase. Statistical distribution-based data reduction techniques have shown promise in this area. Dutta et al. [4], [29] developed end-to-end *in situ* to post-hoc-capable flexible data summarization techniques using Gaussian mixture model-based data representations. Wang et al. [6], [30] also used distributions and added spatial distributions in the analysis framework for accurate data recovery. *In situ*-generated histograms were used by Ye et al. [31] for accelerated post hoc data query-based visual analysis. Another emerging approach for data summarization for efficient post hoc visualization is Cinema [32], which creates an image-based database containing high-resolution data images spanning across various rendering and data parameters. A similar image-based approach was also proposed by Tikhonova et al. [33], [34], where explorable images were utilized for flexible visual analysis of large data. In situ Sort-And-B-spline Error-bounded Lossy Abatement (ISABELA) [5] of scientific data was proposed by Lakshminarasimhan and others. For a more comprehensive survey on data reduction techniques, readers are further directed to the STAR report [35]. Compared to the aforementioned data modeling-based approaches of data reduction, in this work, we focus on reducing the datasets by preserving a small subset of representative data samples. We show that the sub-sampled data can work as a proxy to the full-resolution raw data, and several visualization tasks can be readily performed directly on the sampled data without any additional post hoc processing.

3 Method

3.1 Overview

In this work, we introduce data-driven sampling approaches for prioritizing and preserving the more likely features of the scientific datasets. Instead of assigning equal importance to all data points, we intend to take a data-driven approach and perform fast global+local data importance computations. For regular grid scalar datasets, we identify value-based importance and local smoothness-based importance measures that can generically prioritize the possible features of the data. By combining these approaches with

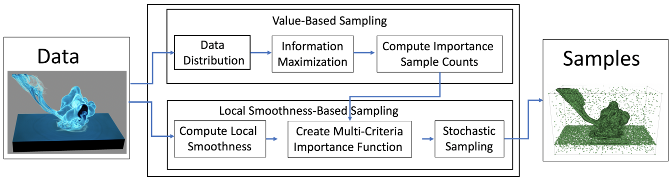


Fig. 1. A schematic workflow diagram of our data-driven sampling approach.

space-filling random sample selection, we come up with a data-driven approach that is fully parallelizable and scalable by design. A schematic of this method is presented in Figure 1.

3.2 Generic Sampling Methods

In this section, we provide a short account of a few existing sampling methods and derive our proposed data-driven sampling approach. For the scope of this paper, we primarily focus on regular grid datasets as inputs to our sampling algorithms.

3.2.1 Simple Random Sampling

Simple random sampling is a popular choice among scientists for drawing samples from unknown populations. The resulting samples generally capture the original data distribution quite well. Given a regular grid dataset and a user-given sampling fraction (say η, where $0 < \eta < 1$), simple random sampling can be performed independently on each grid point. At each grid, a random number (say r, where $0 \leq r \leq 1$) can be generated from a uniform distribution $U \sim unif(0, 1)$, and if $r < \eta$, then this grid point will be accepted. This method ensures each data point has an equal chance of getting selected and does not assign priority to any specific data value that might be of importance to the scientists. One of the popular variants of this method is stratified random sampling, where the data is first divided into strata (groups) and then simple random sampling is applied from within each stratum. An adaptation of this method was proposed recently by Tzu-Hsuan et al. [16], where Shannon's entropy was used to allocate the number of samples within each stratum.

An illustrative example is shown in Figure 2. Here we are using the v02 (water fraction) variable of the asteroid impact dataset. (Details about this dataset are provided in Section 5.1.2.) Figure 2(a) shows the volume rendered visualization of water fraction field representing the plume after the asteroid has impacted the water surface. Figure 2(b) shows the corresponding histogram of this variable. Using simple random sampling, if we assume sampling fraction $\eta = 0.02$ (that is, we want to keep 2% of the original samples), the resulting samples will have a histogram similar to the one depicted in Figure 2(d). As can be seen, these samples are representative of the original data if the resulting data distribution is considered.

3.2.2 Regular Sampling

Another well-known and commonly used sampling method is regular sampling, which allows systematic sample selection. This method sub-samples the data by regularly selecting data points using a predefined interval. Similar to simple random sampling, this method mostly preserves the overall data distribution and yields statistics similar to the original data. Compared to simple random sampling, regular sampling shows sampling artifacts due to the regular nature of sample selection. Since, even after sampling, regular sampling keeps the regular grid structure, the sampled data output is still a regular grid and selected samples do not require the location information.

3.3 Proposed Multi-Criteria Importance-Based Sampling

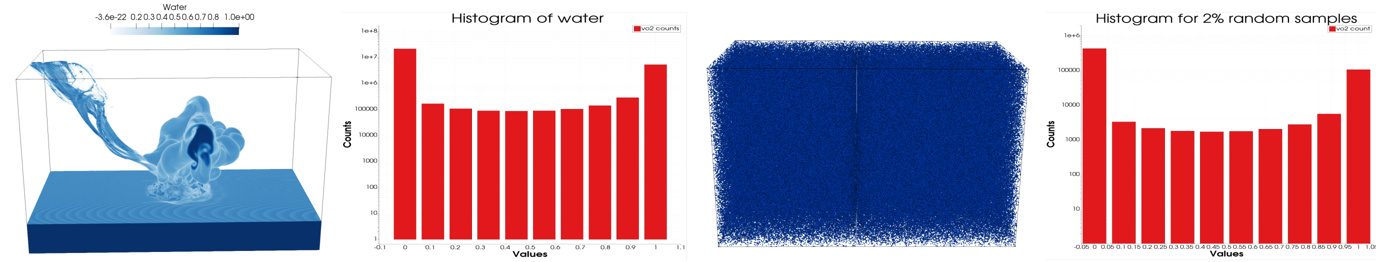
Data analysis and visualization tasks on scientific datasets are often based on the notion of features. Features of a dataset are often regarded as more $important$ for the domain scientists, who often look for those regions of interest while performing query-driven analysis and visualization. Thus, from the visualization aspect, all of the data points of a scientific simulation output are not equally important. When large-scale simulations need to be sampled down to a given storage constraint, it is critical to assign more importance to the feature-like regions.

Formally, given a dataset, we want to formulate an $importance\ function$ I_F that predicts the importance of a data point. If I_F can be constructed such that $0 \leq I_F \leq 1$, then our importance-based sampling technique reduces to generating a random number η at each data point (p_i) and accepting the point if $\eta < I_F(p_i)$. To be applicable across multiple scientific simulations and for $in\ situ$ use, I_F should ideally be constructed automatically and based on only the data properties. Next, we discuss different aspects of importance to effectively create a multi-criteria importance function I_F.

3.3.1 Value-Based Importance Sampling

Problem Formulation: For a scalar variable in a scientific dataset, one important factor in deciding the saliency of a point is its field value [8], [9], [36]. This notion is also observed in the query-driven visualizations where scientists generally want to ask for specific scalar value ranges. Often the important scalar values are those that have a low probability of occurrence. For example, in an image, often background pixels that are abundant are much less important compared to foreground pixels, which are almost always much more rare in the image. The concept of rare values being more important in a dataset has previously been used in visualization literature for data fusion [8], [37], data selection [7], and so on. We adopt a similar approach here. For generating value-based importance sampling, we create our importance function I_F such that the data points whose field values are highly likely in the dataset are assigned low priority. Similarly, the data points with rare or unlikely field values are treated as more valuable and more likely to be of interest to the end users.

Statistical Background and Motivation: Our guiding assumption is that rare values are more likely to be interesting for visualization and discovery. Therefore, we want to choose a sample that overrepresents rare values without completely ignoring the more common values. Let $h(x)$ denote the probability density function of our data, where



(a) Visualization of asteroid impact data (water variable).

(b) Original data histogram.

(c) 2% random sampling output.

(d) Histogram resulting from 2% random sampling.

Fig. 2. Illustration of random sampling on the asteroid impact dataset.

x describes the value of the data. This function is low for rare values and high for common values. Note that this description of the data ignores spatial properties and considers only the data values. By forcing our sampled output to follow a uniform distribution over x with lower and upper bounds denoted by ℓ and u, we naturally up-weigh the rare values and down-weigh the common ones.

To obtain a uniform sample from the data, our approach resembles the rejection sampling algorithm [38]. Rejection sampling is a method for generating a sample from a distribution with density $f(x)$ using a sample from another distribution with density $h(x)$ that is easier to sample. To generate a sample, first a sample is drawn from the distribution with density $h(x)$. Then, a sample u is drawn from the $unif(0,1)$ distribution. If $u \leq f(x)/(C \times h(x))$, then the sampled point is accepted, otherwise it is discarded. C is some number such that $f(x) \leq C \times h(x)$ for all x. The algorithm described below is a simplification of the basic rejection sampling algorithm.

In our case, $h(x)$ is the density of our original dataset. In essence, we approximate both the distribution $h(x)$ and our target uniform using a histogram. Then, fixing $C = 1$ provides the desired sampling of the bins from $h(x)$ to fill the corresponding bins in the target uniform.

Our approach is also closely related to importance sampling [38], [39]. Importance sampling is a Monte Carlo method for computing statistical expectations of one distribution by sampling from another more convenient distribution. Assume that we wish to compute the expectation of a function $g(x)$ with respect to some distribution with density $f(x)$ and we can sample from some other distribution with density $h(x)$. We have

$$\begin{aligned} E_f[g(x)] &= \int g(x)f(x)dx = \int g(x)\tfrac{f(x)}{h(x)}h(x)dx \\ &= \int g(x)w(x)h(x)dx = E_h[g(x)w(x)]. \end{aligned} \quad (1)$$

The expectation that we want using density $f(x)$ is equal to a different expectation using density $h(x)$ with a particular weighting scheme. The weights involve the ratio of the density that we care about to the density that we can use.

In our case, the convenient distribution to sample from will be the original data, which has density function $h(x)$. Using our full original data, we can approximate expectations with respect to $h(x)$. A uniform distribution over the range (ℓ, u) is uniquely defined by its cumulative distribution function $P(X \leq x) = \frac{x-\ell}{u-\ell}$ for $\ell \leq x \leq u$, which has density $g(x) = \frac{1}{u-\ell}$. This can also be written as the expectation of an indicator function:

$$P(X \leq x) = E[\mathbb{1}\{X < x\}] = \int_{\ell}^{x} \frac{1}{u-\ell} dx'. \quad (2)$$

Using these derivations and the theory of importance sampling, we can now estimate the desired distribution. We want a set of weights such that

$$\frac{x-\ell}{u-\ell} = E_h[\mathbb{1}\{X \leq x\}w(x)] = \int_{\ell}^{x} w(x')h(x')dx'. \quad (3)$$

From this, it's clear that

$$w(x) = \frac{1}{(u-\ell)h(x)} \propto \frac{1}{h(x)} \quad (4)$$

satisfies this constraint. Our previously mentioned importance function I_F is essentially a formulation of this weight function w. Therefore, using this I_F, we can select points from our original dataset with probabilities proportional to the inverse of their density, and the resulting samples will be approximately uniform. In practice, we will need to approximate $h(x)$, which can be achieved with a histogram.

Illustrative Example: Let us assume the data histogram is H, where $H(p_i)$ returns the total count of scalar field values close to the value at location p_i. Then,

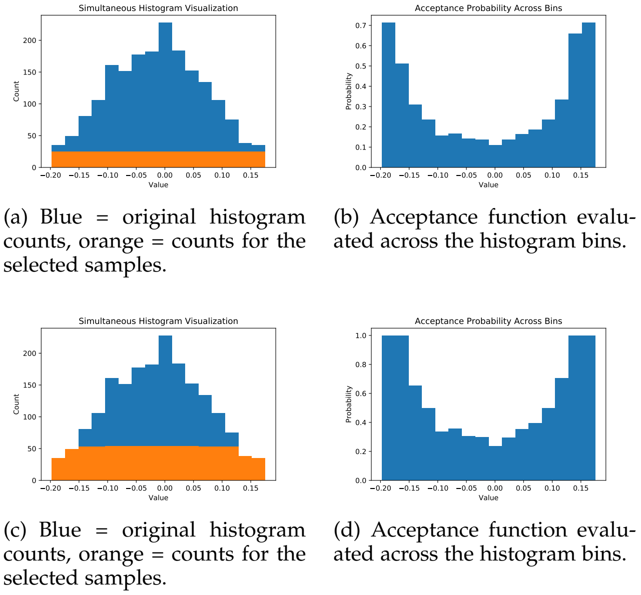
$$I_F(p_i) \propto \frac{1}{H(p_i)} = \frac{C}{H(p_i)}, \quad (5)$$

where C is the proportionality constant; that is,

$$I_F(p_i) \times H(p_i) = C. \quad (6)$$

This is similar to creating a new histogram $H_{Samp}(p_i) = I_F(p_i) \times H(p_i)$, whose counts are equal across all bins ($= C$); that is, from each histogram bin of H, we need to collect C samples to achieve our value-based importance sampling. This construct, in turn, results in entropy maximization via sampling because the histogram of the sampled data will be as uniform as possible.

Now, given the sampling ratio η and total number of data points in the dataset N, if there are B bins in histograms H and H_{samp}, then $B \times C = N \times \eta$; that is, $C = (N \times \eta)/B$. Based on the user-given parameters η and B, if C is smaller than the smallest count across all bins of H for the input dataset, then this algorithm will simply need to pick C samples from each bin of H with $I_F(p_i) = \frac{C}{H(p_i)}$. For illustration, this is shown in Figure 3(a), where the data is generated from a Gaussian distribution.



(a) Blue = original histogram counts, orange = counts for the selected samples.

(b) Acceptance function evaluated across the histogram bins.

(c) Blue = original histogram counts, orange = counts for the selected samples.

(d) Acceptance function evaluated across the histogram bins.

Fig. 3. Illustration of value-based importance sampling. (a) Situation when sampling ratio $\eta = 0.2$; (b) situation when sampling ratio $\eta = 0.4$.

For sampling ratio $\eta = 0.2$, we can collect $C = 24$ samples across all $B = 16$ bins of this histogram since the lowest count across all the bins is 35. For this case, the acceptance function I_F can be computed to prioritize the rare values over the more likely ones, as shown in Figure 3(b).

For handling the scenario where C is larger than the smallest count C_{lowest} in H, the sampled histogram will not be uniform since we cannot take C samples across all the bins. This situation will occur if, for example, $\eta = 0.4$ (as in Figure 3). Then, $C = 48$, which is larger than the smallest bin count of $C_{lowest} = 35$ in this example. In this scenario, we can only take C_{lowest} samples from the bin with the smallest count. In fact, for all the bins b_js with corresponding counts c_js, where $c_j < C$, we can only take c_j samples. In order to still pick user-given $M = N \times \eta$ samples, we need to distribute the differences $D = \sum C - c_j$ (where $c_j < C$) to the bins b_js with counts c_j, where $c_j > C$. This is shown in Figure 3(c). Now the resulting histogram in orange is not fully uniform, but it is as uniform as possible given the inputs. The corresponding acceptance function is shown in Figure 3(d), which again illustrates the concept of selecting rare values with a higher chance.

Value-Based Sampling Algorithm: The final algorithm, Algorithm 1, for achieving value-based importance sampling begins by creating the histogram of the input dataset. Next, we sort the histogram bins according to their counts from smallest to highest. Starting from the first bin (with the smallest count), we assign target samples to be picked as $min(c_j, C)$, that is, the minimum of the current bin count (c_j) or the current target samples (C) for each bin. If $c_j < C$, then C is updated by computing the remaining samples to be picked with the remaining number of bins. We continue this process until we exhaust all the bins or we reach a bin where $c_j > C$. Starting from this bin, since all the remaining bins will satisfy the property $c_j > C$ (we are working on a sorted histogram), all the remaining bins will get assigned current C samples. After creating the target histogram counts, we perform a bin-wise division of the count values between the original and target histogram. This will give us our importance function I_F. Using this, we can now perform data sampling. For each point p_i in the dataset, we compute a random number between 0 and 1 and compare with $I_F(p_i)$. If the random number is lower than $I_F(p_i)$, then p_i is accepted.

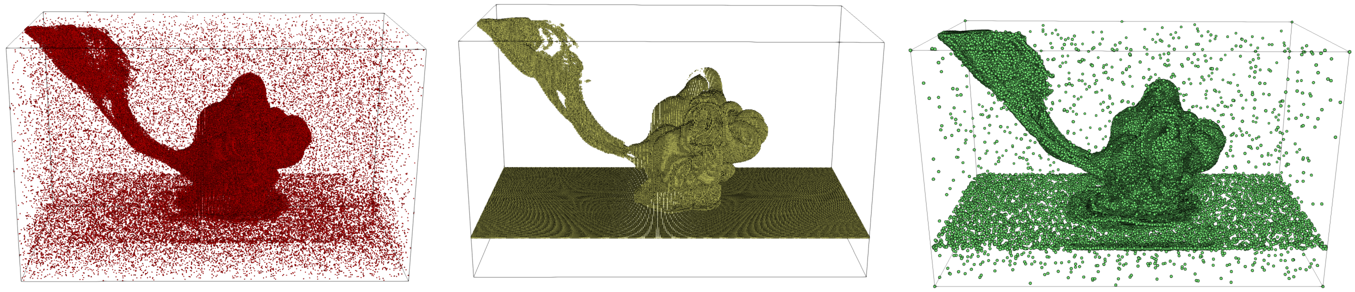
Algorithm 1: Importance histogram creation for value-based sampling technique.

```
Input: $D$ (data), $N$ (number of data points), $M$
  (number of samples), $B$ (number of bins)
Result: $I_F$ (importance function/histogram for
        selecting $M$ samples from $N$ data points)

$H \leftarrow$ histogram($D$, $N$, $B$);
$H \leftarrow$ sortAscending($H$);
$I_F \leftarrow$ zeros($B$);
$C \leftarrow M/B$ ; // Expected number of samples
$j = 0$;
while $j < B$ do
    $c_j \leftarrow H[j]$ ;          // Count in bin j
    if $c_j < C$ then
        $I_F[j] = c_j$;
        $M \leftarrow M - c_j$;
        $B \leftarrow B - 1$;
        $C \leftarrow M/B$;
        $j = j + 1$;
    else
        for $k$ to $B$ by 1 do
            $I_F[k] \leftarrow C$;
        end
        break
    end
end

/* Normalize by histogram count        */
for $j \leftarrow 0$ to $B$ by 1 do
    $I_F[j] \leftarrow I_F[j]/H[j]$;
end
```

We can refer to Figure 4 for comparing this algorithm's performance. Compared to the random sampling that would distribute the samples in a space-filling manner throughout the space, value-based importance sampling provides more samples from the feature region of the asteroid impact dataset.

3.3.2 Smoothness-Based Importance Sampling

Problem Formulation and Motivation: When analyzing a data point in a scientific dataset, an important aspect is its local smoothness or homogeneity. The following have been used previously: homogeneity for clustering [40], data reduction using modeling [29], and bit-map index-based compression [16]. In the context of data sampling, if field values change abruptly in a local region, then this region requires more representative samples to perform reconstruction or local property prediction. That is, the points in this region will be probabilistically assigned higher importance. This notion is different from the above mentioned value-based importance, as the importance of a given location is

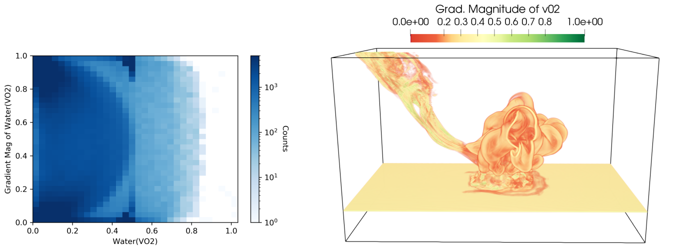


(a) Value-based importance samples.

(b) Local smoothness-based samples.

(c) Joint multi-criteria based samples.

Fig. 4. Comparison of the different sampling algorithms for a sampling ratio of 2%. (a) value-based sampling, (b) local smoothness-based sampling, and (c) joint multi-criteria sampling. Compared to the sampling output from the random sampling method as given in 2(c), these data-driven methods reveal the features of the data much better.



(a) 2D histogram of water fraction and its gradient magnitude.

(b) Volume visualization of gradient of water fraction.

Fig. 5. Motivation for use of gradient for sampling. From asteroid impact data, (a) 2D histogram of field value vs. gradient. It shows that a given field value can have varying gradient based on its spatial location, (b) high gradient regions correlating with interesting parts of the data.

not dependent on the scalar value of this point, rather on how fast the local field values are changing. Depending on its location, the two points with same field values can have different degrees of representativeness; in one region, since the values change slowly, this data point may be assigned lower importance for the use cases like post hoc reconstruction, etc. The other data point with the same field value in a high gradient region will be assigned higher priority. This example in Figure 5(a) shows a 2D histogram of the water variable and its corresponding gradient magnitudes from the Asteroid dataset. It is evident that the same field values can have varying gradient.

As a generic measure of importance, gradients can be effectively used for selecting interesting sample points. High gradient regions are often of interest to the domain experts and they often identify regions of feature. Previously, gradients have been utilized for automatic detection of boundaries and generating color-maps, e.g. in [41], [42]. Figure 5(b) shows the gradient magnitude of the water variable from the asteroid dataset, and evidently the interesting phenomena of water splash due to asteroid impact is well characterized by the gradient field. Also from the point of view of data reconstruction, high gradient regions are of high importance because reconstruction is much harder in those regions. Thus, for gradient-based sampling, our importance function I_F is directly proportionate to the gradient; that is, for a point p_i with gradient function G, $I_F(p_i) \propto G(p_i)^k$, where $k \in \mathbb{R}$. Here, k is a parameter that decides how strongly the gradient will influence the sampling result. For example, if $k \rightarrow \infty$, then we will start collecting samples that have the highest gradient, without any randomness in the selection method.

Smoothness-Based Sampling Algorithm: For achieving the above-mentioned gradient-based sampling, we proceed by creating a histogram of gradients with a user-given bin number. Given $M = N \times \eta$ samples to be picked, we start to assign the samples from the bin with the highest count. We continue this process until all the samples are picked. Now the resulting histogram can be bin-wise divided with the original gradient histogram to obtain the bin-wise acceptance probability. Since we know which bin each data point belongs to based on the gradient, we know which data point has what chance of being accepted. Now for each data point, a random number between 0 and 1 is generated and is compared with the corresponding acceptance probability for acceptance. Algorithm 2 shows this process.

Algorithm 2: Importance histogram creation for smoothness-based sampling technique.

```
Input: $D$ (data), $N$ (number of data points), $M$
  (number of samples), $B$ (number of bins)
Result: $I_F$ (importance function/histogram for
        selecting $M$ samples from $N$ data points)
$G \leftarrow$ computeGradient($D$);
$G_{mag} \leftarrow$ computeGradientMagnitude($G$);
$H \leftarrow$ histogram($G_{mag}$, $N$, $B$);
$I_F \leftarrow$ zeros($B$);
$C \leftarrow M/B$ ; // Expected number of samples
$j = B - 1$;
while $j >= 0$ and $M > 0$ do
    $c_j \leftarrow H[j]$ ;            // Count in bin j
    $I_F[j] = c_j$;
    $M = M - c_j$;
    $j = j - 1$;
end
/* Normalize by histogram count        */
for $j \leftarrow 0$ to $B$ by 1 do
    $I_F[j] \leftarrow I_F[j]/H[j]$;
end
```

The result of the above-mentioned gradient-based sampling is shown in Figure 4(b). Compared to 4(a), we can see that more samples have been allotted to the asteroid impact region. Given the same storage constraint, it is noteworthy that this gradient-based sampling does not have the space-filling property compared to random sampling and value-based sampling.

3.4 Fused Sampling Method

All the algorithms discussed above have their own benefits and drawbacks. In this section, we discuss a workflow that takes advantage of the positives of these generic algorithms by combining them in an efficient way. The value-based sampling attempts to prioritize the samples that are primarily rare and possibly of interest to the users. The local smoothness-based algorithm attempts to select samples that are in non-homogeneous spatial regions and likely to be of interest. Random sampling has the property of filling up the 3D space because it treats all the points equally. Uniform sampling selects the data at regular intervals, which can be useful if users want to perform reconstruction of a region from the point samples. If linear interpolation-based reconstruction needs to be performed, uniform sampling can ensure the tetrahedrons from the sampled data mesh are not ill-shaped.

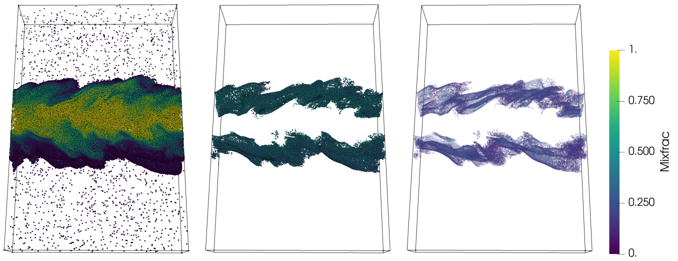
3.4.1 Joint Multi-Criteria Sampling

To perform the joint multi-criteria sampling based on the previous approaches, we prioritize the feature-based algorithms. Similar to the value-based method, here we also first create the histogram for expected samples to compute our importance function. We start by creating the 1D value-based importance function as described in Section 3.3.1. After knowing the target number of samples for each value bin, the gradient information is used for selecting samples that fall into the same value bin. Using a 2D histogram with field values and their corresponding gradients, our goal is to select the samples required by the value-based sampling scheme by prioritizing their gradients. Recall that the importance function for gradient has a parameter k. As $k \to \infty$, we basically take the samples starting from the highest gradient bin. If $0 < k < \infty$, we again attempt to assign the samples proportionate to $G(p_i)^k$. Since the highest gradient bins for a given value bin may not always have enough bin count to provide all the samples as requested, there are leftover samples. We attempt to equally distribute the remaining samples to all the bins, which essentially amounts to the random sampling behavior and provides samples from the empty regions. To achieve this last stage, we employ a sample assignment technique similar to the sorted histogram approach of Section 3.3.1.

The output of this algorithm is shown in Figure 4(c). For the same storage (2%), we can compare this output with only value-based (Figure 4(a)) and only gradient-based (Figure 4(b)) methods—the combined algorithm retains the space coverage and provides more samples from the region of interest.

3.4.2 Combined Independent Sampling

Although we recommend the use of our joint sampling method (mentioned above), in our workflow we allow users to create their own combined sampling strategies. For each of the independent algorithms (random, uniform, value-based, gradient-based), we take a tuple $w_i, i = 1, ...4$ as input from the users, and this is used as a weight factor for each algorithm. With the user-given overall sampling ratio η, samples are collected by running each algorithm where the corresponding sampling ratio $\eta_i = \eta \times \frac{w_i}{\sum w_i}$. Correctly setting the w_is is crucial in this mode and requires much detailed understanding of the simulation data properties. For selecting the optimal weights for the in situ use-case, the users are advised to first tune the parameters on a post-hoc workflow with smaller scale data. Such tuned parameters can be transferred to the in situ run.



(a) 5% samples. (b) Queried points. (c) Volume rendering. (d)

Fig. 6. Sample-based query-driven visualization of combustion dataset on Mixfrac field. Sampling ratio 0.05 is used. Result of an example query of Mixfrac $\geqslant$ 0.3 AND $\leqslant$ 0.5 is shown. Figure 6(a) shows the initial sample points selected. Figure 6(b) shows the points that satisfied the query. Figure 6(c) provides a volume-splat-based visualization of the query results.

4 Sample-based Visualization Approaches

4.1 Query-Driven Visualization

Query-driven visualization (QDV) is an effective way to discover and understand features in scientific datasets [13], [14]. QDV is well-known because it reduces the computation workload and helps the application experts focus on regions of interest. In this work, to facilitate experts with query-driven visualization and analysis capability, we incorporate QDV on the sampled output of the datasets. By directly querying the sampled output, scientists can quickly assess the result of the query and visualize it. However, since the query is performed on the sampled data, the result obtained is only an approximated result of the queried result. Once the scientists find a suitable query range by analyzing the sampled data, further refinement of the query region can be done by reconstructing the required regions of the data to its original full resolution. The techniques used for reconstruction are discussed in Section 4.2.

Given a specific query, for example, the value of variable $X \geq$ a AND $X \leq$ b, we first isolate the sample points that satisfy this query. Now, these point samples can be directly visualized for analysis. To enhance the quality of the visualization, in this work, we also employ a volume splat-based visualization strategy. Essentially, we construct a new scalar field, where the grid points that satisfied the query contain the true scalar value of the point and then each of these points also splats a small contribution to its neighboring grid points. Given that for any specific neighboring grid point, multiple samples can splat contribution to it, the average of all the contribution values is assigned as the final value of that specific neighboring grid point. When assigning contributions to a neighboring grid point, the contribution falls off proportionately with the inverse of the square of the distance between the neighboring point and the contributing point. Hence, if v is the scalar value of a contributing point P, and d is the distance between point P and its neighbor Q, then the scalar value splatted to Q by P is v/d^2. Finally, if Q has n neighboring points that can contribute partially to it, then final value $v(Q)$ at the point Q is computed as

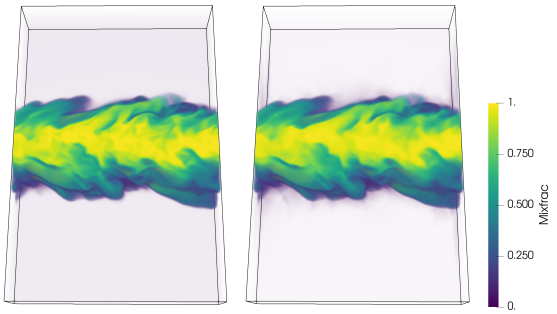
$$v(Q) = \frac{\sum_{i=0}^{n}(v(P_i)/d_i^2)}{n}. \tag{7}$$

This new scalar field is then visualized by a volume rendering technique for visually exploring the results of the user-specified query in the spatial domain. Since this QDV is performed directly on the sampled dataset, the operation is fast and the experts can visually explore the results almost interactively.

In Figure 6, we demonstrate the usefulness of the QDV using the Mixfrac field of combustion dataset. This dataset is a turbulent simulation, and the Mixfrac variable denotes the proportion of fuel and oxidizer mass. This value generally provides the location of the flame where the chemical reaction rate exceeds the turbulent mixing rate [8], [43]. The spatial resolution of this dataset is $480 \times 720 \times 120$. Since it was previously observed that the Mixfrac values around 0.42 represent the frame region [8], [43], we performed a query of Mixfrac values $\geq$ 0.3 AND $\leq$ 0.5 on our sampled dataset. In Figure 6(a), we show the 5% sample points from the original dataset for Mixfrac variables using our proposed method, and Figure 6(b) highlights the points that satisfy the above query when applied on our 5% samples. A simple point rendering is used in this figure, where each point is represented using a sphere glyph. We can see a dense set of points were selected as a result of the query, demonstrating that our sampling scheme is able to keep more sample points from the important feature regions of the data. Figure 6(c) provides the volume splat view of the queried results as another form of visualization.

4.2 Reconstruction-Based Visualization

In order to visualize and analyze the data in its entirety at full resolution, we also enable data reconstruction from the stored sampled dataset. A naive way of reconstructing the whole data would be to perform nearest neighbor interpolation using the sampled point set. Besides nearest neighbor interpolation-based reconstruction, we also provide a linear interpolation-based reconstruction technique. Since the nearest neighbor-based technique is faster compared to linear interpolation, for very large-scale datasets, it can be used to quickly reconstruct the data. However, for a higher quality of visualization, the linear interpolation scheme is used. Note that by paying more computational cost, higher-order interpolations can also be used to increase the quality of the reconstruction further.



(a) Original data. (b) Reconstructed data from 5% samples. (c)

Fig. 7. Volume visualization of Mixfrac field of combustion dataset. Figure 7(a) shows the original data rendering, and Figure 7(b) provides the reconstructed data rendering from 5% sample points. Linear interpolation-based reconstruction is used in this example.

To linearly interpolate the data, first, a 3D convex hull is computed using all the sampled points. Then, the points are converted to a polygonal mesh using 3D Delaunay triangulation. Next, for each grid point in the reconstruction grid, the value is obtained by linearly interpolating scalar values from the vertices of the simplex that encloses the current grid point. To ensure that reconstructed and original volumes match, the boundary points (8 points for the 3D volume) are also added to the sampled point set prior to applying reconstruction. Once the reconstruction is complete, we allow traditional volume rendering and isocontour-based visualizations for the exploration of the dataset.

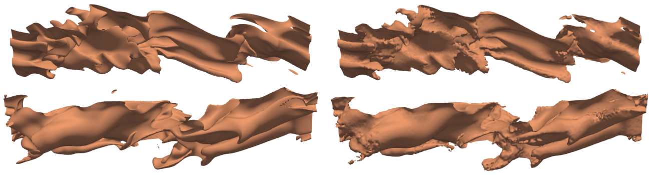
Volume-Based Visualization: We employ traditional ray casting-based techniques for volume-based visualization of the reconstructed data. The users can modify the transfer function as necessary to explore features in the reconstructed volume. In Figure 7, we show the volume-based visualizations of the Mixfrac field of combustion dataset. Figure 7(a) shows the volume rendering of the full resolution original data, and Figure 7(b) shows volume rendering of the reconstructed Mixfrac field. It is observed that the reconstructed field is visually very similar to the original data.

Isocontour-Based Visualization: We also facilitate isocontour visualization on the reconstructed data. Users can specify feature-specific isovalues to render isosurfaces and visualize them interactively. We demonstrate the isocontour visualization in Figure 8, where the isocontour of Mixfrac = 0.42 is shown both from the original data (Figure 8(a)) and the reconstructed data (Figure 8(b)) from 5% samples. The isovalue of 0.42 is important in this dataset because this value of Mixfrac represents the flame structure of the turbulent combustion process. By visually comparing the two contours in Figure 8, it can be concluded that the proposed sampling method is able to preserve the global structure of important features in data with high accuracy even at low sampling rates (in this case only 5% samples were taken).

5 Case Study and Evaluation

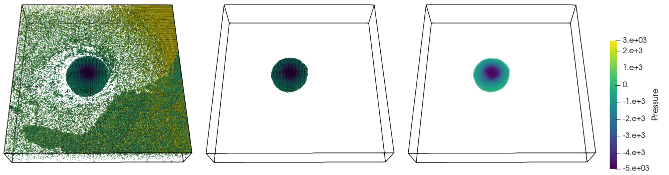
5.1 Case Study

In the previous section, we provided the results from the combustion dataset. Here we further detail two other



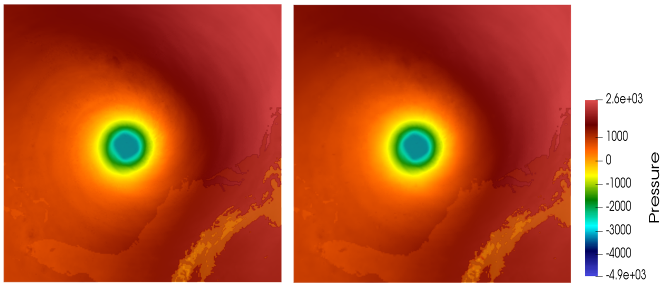
(a) Isocontour on original data. (b) Isocontour on reconstructed data from 5% samples.

Fig. 8. Isocontour visualization of Mixfrac = 0.42 of combustion dataset. Figure 8(a) shows the pressure isocontour extracted from the original data, and Figure 8(b) depicts the same isocontour extracted from the reconstructed field (reconstructed multiplied using 5% samples).



(a) 5% samples. (b) Queried points. (c) Volume rendering. (d)

Fig. 9. Sample-based query-driven visualization of Hurricane Isabel dataset on pressure field. Sampling ratio 0.05 is used. Result of an example query of pressure $\geq$ -5000.0 AND $\leq$ -500.0 is shown. Figure 9(a) shows the initial sample points selected. Figure 9(b) shows the points that satisfied the query. Figure 9(c) provides a volume-splat based visualization of the query results.



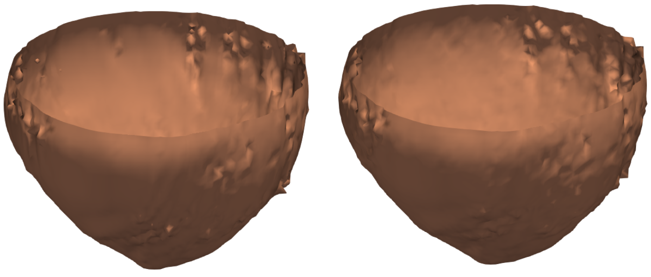
(a) Original data. (b) Reconstructed data from 5% samples. (c)

Fig. 10. Volume visualization of pressure field of Hurricane Isabel dataset. (a) Shows the original data rendering, and (b) provides the reconstructed data rendering from 5% samples. Linear interpolation-based reconstruction is used in this example.

datasets: Hurricane Isabel and Asteroid Impact.

5.1.1 Hurricane Isabel Data

The Hurricane Isabel dataset was used to study the impact of the hurricane Isabel. The dataset is courtesy of NCAR and the U.S. National Science Foundation (NSF). The dataset was created using the Weather Research and Forecast (WRF) model. The spatial resolution of the data is 250 × 250 × 50. In this dataset, the pressure variable is one of the most important because this variable can represent the eye of the hurricane, and typically scientists isolate the eye of the storm by querying the low-pressure regions [44]. To study the effectiveness of the proposed sampling scheme, we used the pressure field from this dataset and performed a query on low pressure $\geq$ -5000.0 AND $\leq$ -500.0 on the sampled dataset. Figure 9(a) shows all the sampled points selected by the proposed sampling scheme, and in Figure 9(b) we show the sample points that satisfied the above query. Figure 9(c) provides the volume splat-based visualization of the queried points. It can be seen that by analyzing the sampled points directly, the proposed method is able to preserve the hurricane eye feature quite well.



(a) Isocontour on original data. (b) Isocontour on reconstructed data from 5% samples.

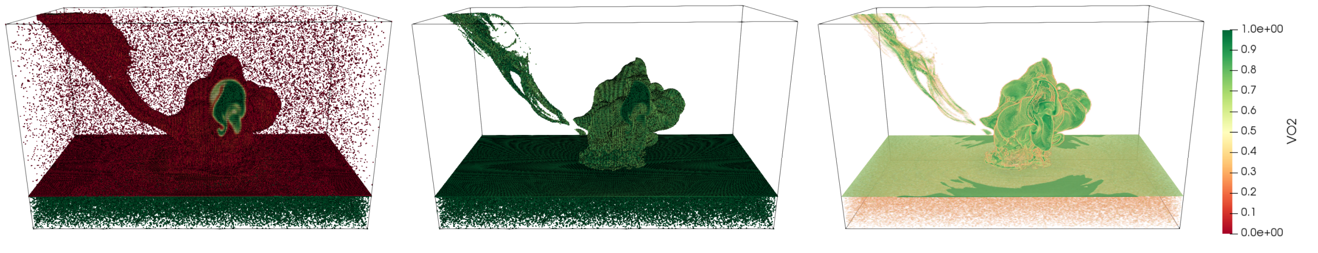
Fig. 11. Isocontour visualization of pressure = -500 of Hurricane Isabel dataset. Figure 11(a) shows the pressure isocontour extracted from the data, and Figure 11(b) depicts the same isocontour extracted from the reconstructed field (reconstructed using 5% samples).

To then visually compare the result of the reconstructed data to the original data, we used volume rendering. The rendering results are shown in Figure 10, where Figure 10(a) shows the rendering of the original data and Figure 10 provides a rendering of the reconstructed data using the same rendering parameters. We can see that the reconstructed data from the sampled points produce a smooth visualization and is visually very similar to the original data. In Figure 11, we provide the isocontour-based visualization. Isocontour of pressure = -500.0 is extracted from both the original data (11(a)) and the reconstructed data (11(b)). It is observed that the isocontour extracted from the reconstructed data matches well with the true isocontour.

5.1.2 Asteroid Impact Data

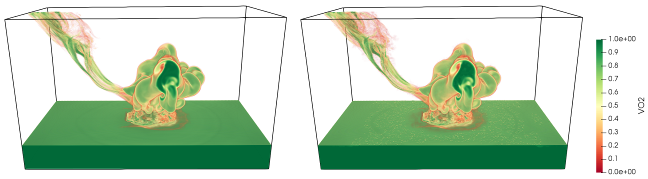
The Deep Water Impact Ensemble dataset [45] represents an ensemble of simulations run at Los Alamos National Laboratory to study Asteroid Generated Tsunami, or AGT. To evaluate our sampling scheme, we used one of the ensemble members in this work where the spatial resolution of the data is 300 × 300 × 300. We used the volume fraction of water variable, denoted by v02, to conduct our study. By studying the v02 variable, the splash of the water into the atmosphere can be visualized. The value of v02 lies between 0.0 and 1.0, where 1.0 means pure water. This enables the study of ablation and ejecta material as the asteroid enters and subsequently impacts the water, sending a plume of material into the surrounding area and up into the atmosphere [46], [47].

To visualize the v02 variable where the values of water fraction are high, we performed a query on the sampled dataset where v02 $\geq$ 0.75. Figure 12(a) shows the sample points initially selected by the proposed sampling algorithm, and Figure 12(b) provides the result of the query through direct point rendering. The splat volume-based visualization of the queried result is shown in Figure 12(c), and the span of the area where the water splashed after the asteroid impact can be observed.



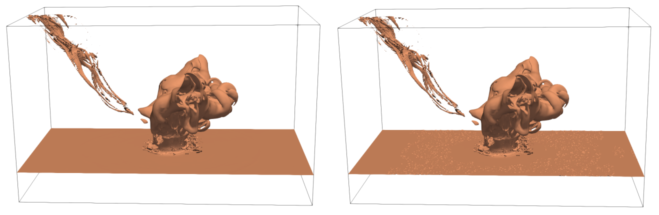
(a) 6% samples. (b) Queried points. (c) Volume rendering. (d)

Fig. 12. Sample-based query-driven visualization of asteroid dataset on v02 field. Sampling ratio 0.06 is used. Result of an example query of v02 $\geq$ 0.75 is shown. Figure 12(a) shows the initial sample points selected. Figure 12(b) shows the points that satisfied the query. Figure 12(c) provides a volume-splat-based visualization of the query results.



(a) Original data. (b) Reconstructed data from 6% samples. (c)

Fig. 13. Volume visualization of v02 field of asteroid dataset. (a) Shows the original data rendering, and (b) provides the reconstructed data rendering from 6% samples. Linear interpolation-based reconstruction is used in this example.



(a) Isocontour on original data. (b) Isocontour on reconstructed data from 6% samples.

Fig. 14. Isocontour visualization of v02 = 0.8 of asteroid dataset. (a) Shows the v02 isocontour extracted from the data, and (b) depicts the same isocontour extracted from the reconstructed field (reconstructed using 6% samples).

The volume rendering results on the reconstructed v02 field of the asteroid data are provided in Figure 13(a). Comparing the image in Figure 13 (which was generated from the original data), we can see that the reconstructed v02 field from the sampled points is able to produce visually pleasing and accurate volume visualization for analysis. Isocontour rendering from the reconstructed data is also compared visually with the isocontour extracted from the original data. The results are depicted in Figure 14, where Figure 14(b) presents the isocontour of v02 = 0.8 extracted from the reconstructed data, and the true isocontour from the data is shown in Figure 14(a). From the above results, it can be observed that the sample points selected by the proposed sampling scheme can be used effectively to produce different representations of the data that can be used to answer scientific queries with high accuracy.

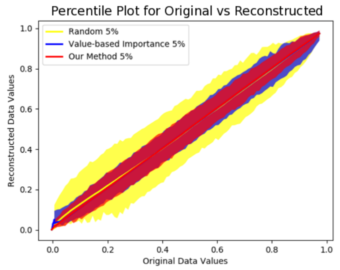


Fig. 15. Scatter plot showing 99 percentile of the original and reconstructed values for three methods: random, value-based, and our method at 5% sampling ratio for combustion dataset.

5.2 Evaluation

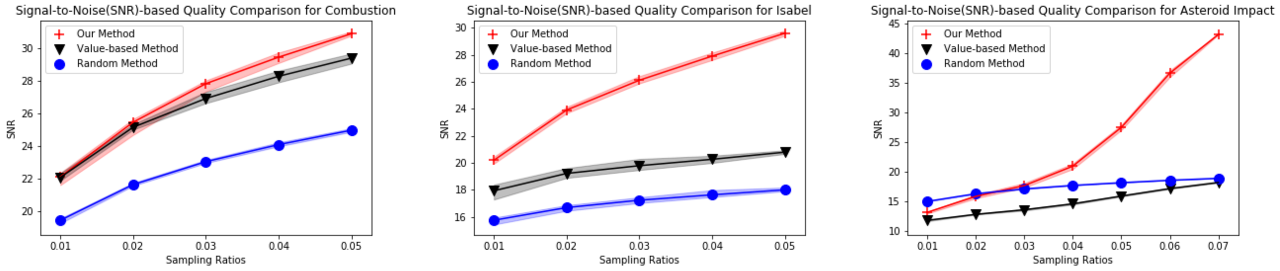
In the large data context, reconstruction is not the primary goal of this paper. For evaluation purposes, we reconstruct the original datasets from the samples coming out of the different sampling methods and study how well the original data properties have been restored. To estimate the quality of the reconstruction we use signal-to-noise ratio (SNR) as one of our quality measures. We define SNR as

$$SNR = 20 * log_{10} \frac{\sigma_{raw}}{\sigma_{noise}}. \quad (8)$$

The key component in this formula is the ratio in the log. The quantity σ_{raw} is the overall standard deviation of the original data. The quantity σ_{noise} is the standard deviation of the error of the reconstruction (the difference between the original data and the reconstruction). As the reconstruction improves, σ_{noise} gets smaller while σ_{raw} remains constant and the SNR increases. Larger values indicate better reconstructions.

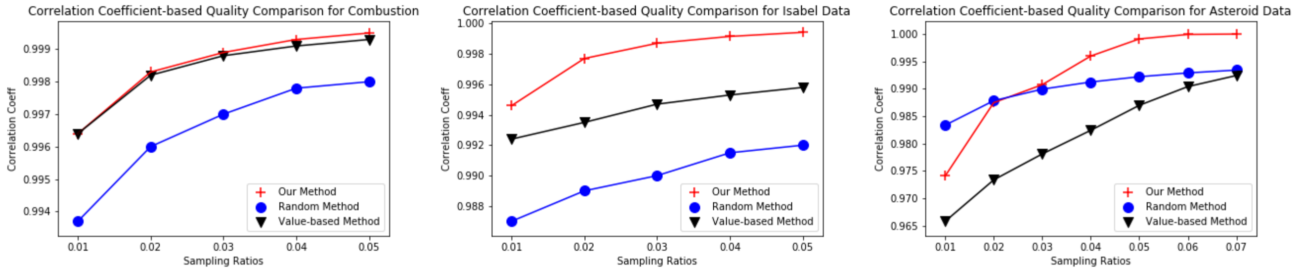
Apart from SNR, we also provide the correlation coefficient between the original and reconstructed datasets. If X and Y are the original and reconstructed datasets, respectively, when a scatter plot is created taking points from each dataset, for reconstruction without error, we expect all the points to lie on the $y = x$ line. For illustration we can refer to Figure 15. This percentile scatter plot depicts the reconstructed values (y axis) for each of the original values (x axis) for the three methods: random, value-based, and our proposed joint criteria-based sampling methods for combustion dataset with sampling rate 5%.

The upper and lower limits were decided by taking the top 99 percent of the absolute difference between the



(a) For Combustion. (b) For Isabel. (c) For Asteroid

Fig. 16. Comparison of SNR across different sampling methods for three different datasets.



(a) For Combustion. (b) For Isabel. (c) For Asteroid

Fig. 17. Comparison of correlation coefficient across different sampling methods for three different datasets.

original value and the reconstructed value at each point in the structured grid, culling the worst 1%. The solid colored lines show the mean value of the reconstructed points. Narrower spreads of color represent better reconstruction. Since ideally all the points should lie on the $y = x$ line, the linear correlation coefficient, in this case, would work as a measure to capture this linear relationship.

For the three datasets—Combustion, Isabel, and Asteroid—Figure 16 shows the SNR for different sampling rates using a default configuration of the sampling methods. (We explore optimizing the number of bins for our method in Section 7.4). We used linear interpolation for comparing the results across the three datasets. All the experiments were run 10 times and the average numbers are reported in the tables. For the charts in Figure 16, we also show the lower bound and upper bound of the SNR to show the spread of the results. From Figure 16, it can be observed that for a given sampling rate, our proposed method generally performs better than the value-based method and random sampling method. For the combustion and Isabel datasets, our proposed method consistently performs better than the other two methods. For asteroid, although initially random sampling performs better, our proposed method reaches much higher SNR. The correlation study for these datasets is provided in Figure 17. In this figure, we provide the correlation coefficient between the original data and each of the reconstructed ones. This figure also shows that our method generally performs best for these selected sample rates.

6 Error Analysis

As we move towards very low sampling rates from large-scale datasets, getting an understanding of the amount of information loss becomes more and more important. In our proposed sampling approach, we provide a local error analysis while the sampling is being performed on the data. Even though our *in situ* scenario generally has a sampling rate that is predefined by a user's given data bandwidth on a supercomputer, this error analysis can also be used for determining the sampling rate.

Since our proposed sampling method assigns more importance to the feature regions and selects more samples from those regions, the original data distribution is not preserved if we simply create a distribution from the resulting samples. Thus, for estimating local information loss in our method, we first resample the sampled output to a regular grid and then we compare with the original data to understand information loss in that local region through local data distributions and point-wise error methods. We are now dealing with small blocks of data, so instead of using a nearest-neighbor interpolation scheme (which is very fast but less accurate), we can employ more sophisticated methods for resampling using Delaunay triangulation first on the point samples and then using linear interpolation within each tetrahedron, as discussed in Section 4.2. Now we get regular grid data out of the samples (D_{samp}) and use this for error measurement with the original data block (D).

For estimating point-wise errors, such as max absolute error, max value range–based relative error [48], or SNR, we resample the sampled output to the original grid locations. Now absolute error E_{abs} can be computed as $E_{abs} = |D - D_{samp}|$, and its maximum value can be used to indicate the max absolute error in that local region. Similarly, relative error E_{rel} is computed as $E_{rel} = |D - D_{samp}|/R$, where R is the range of the data in the local neighborhood, and its maximum can be used to indicate the worst case error

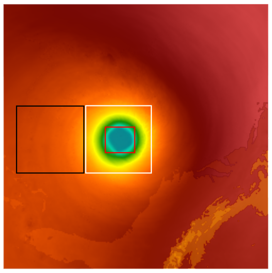
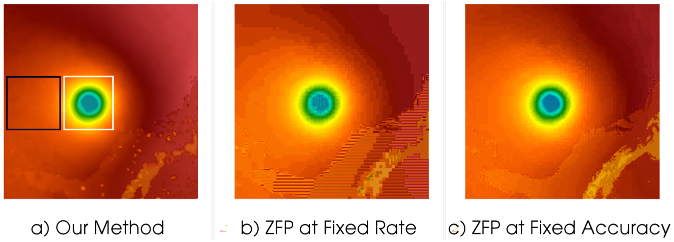


Fig. 18. Local error analysis to be performed (as shown in Table 1) at different regions of Isabel dataset. The white box and the red box show the regions that encapsulate the hurricane eye (feature) at varying size, and the black box shows a non-feature region.



a) Our Method b) ZFP at Fixed Rate c) ZFP at Fixed Accuracy

Fig. 19. Visual comparison with ZFP at around 1% storage. The original dataset is shown in Figure 18. As can be observed, in the feature region (hurricane eye, white box), our method outperforms both ZFP at fixed rate and ZFP at fixed accuracy. As we increase the sampling rate to 5%, with more data, ZFP starts to perform better.

scenario in a relative error sense. SNR is another popular method for quality estimation (as discussed earlier), and it can also be used in this scenario.

For understanding loss in distributional properties, we use methods such as Kullback–Leibler (KL) divergence after computing probability distribution functions (PDFs) P from D and Q from D_{samp}. KL divergence E_{KL} measures the loss in information when Q is used in place of P and is given as

$$E_{KL} = \sum_{x \in X} P(x) log(\frac{P(x)}{Q(x)}). \tag{9}$$

EMD (also known as Wasserstein distance) computes the L1-norm between two cumulative distribution functions (CDFs) generated from P and Q and can be used to understand the amount of cost for changing one CDF to another. For both KL-divergence and EMD, lower values are better with 0 being the lower limit (denoting perfect reconstruction, zero information loss). In our workflow, users can turn on an error flag and provide a desired block size for understanding the local information loss.

In Figure 18, the Isabel dataset is shown where three regions have been selected for the illustration of local error analysis. The region enclosed by the black box is a non-feature region. The white box (larger size) and the red box (smaller size) are shown around the feature of the data (hurricane eye). The result of local error analysis is shown in Table 1 for these three regions. In this case, the value range for this dataset is quite high, (minimum = -4931.54; maximum = 2594.97; range = maximum $-$ minimum = 7526.51). This is the reason for the max absolute errors to appear a little large. When the value range is taken into consideration (as in relative error), it can be observed that the max local error is quite low. It can also be seen from this table that the local errors generally go down as more samples have been selected for all these regions. Also, for a given sampling ratio, feature regions have much tighter local error bounds compared to the non-feature regions, without applying any explicit feature detection.

TABLE 1
Local error analysis for Isabel dataset at varying sampling rates.

Region	Sample Rate	Max. Abs. Error	Max. Rel. Error	KL Dist.	EMD
Red (Feature)	1%	292.3	0.06	0.012	5.5
	3%	244.9	0.05	0.003	2.5
	5%	116.3	0.02	0.003	2.3
White (Feature)	1%	440.9	0.07	0.029	13.3
	3%	440.9	0.07	0.007	6.3
	5%	440.9	0.07	0.003	4.1
Black (Non -feature)	1%	312.7	0.20	0.019	15.1
	3%	302.6	0.20	0.004	5.3
	5%	332.8	0.22	0.002	3.5

TABLE 2
Storage usage at different sampling ratios for Isabel dataset.

Hurricane Isabel Dataset				
Sampling Ratio(%)	Data Size (bytes)	Index Size (bytes)	Total Size (bytes)	Effective Sampling Ratio(%)
0.5%	62628	16308	78936	0.63%
1%	125100	27744	152844	1.2%
2%	249208	49616	298824	2.3%
3%	374492	70324	444816	3.5%
4%	498488	90624	589112	4.7%
5%	624276	109956	734232	5.8%

7 Discussion

7.1 Storage Handling

Our method for sampling scientific datasets is primarily intended for use where a storage constraint (i.e., how much data can be stored) is pre-specified. We work with regular structured grid datasets that have implicit point locations. We create point samples that require the storage of explicit point locations in addition to the sampled field values. The naive representation stores point locations as (x_i, y_i, z_i) tuples. Our first optimization simply stores these tuples with their corresponding integer indices I_i, where $I_i = x_i + y_i * XDIM + z_i * XDIM * YDIM$ from the original structured representation. So, instead of storing three floats, we now store one integer index for each point.

In the next stage, to further optimize the storage needed for the indices, we observe that the point indices are a highly compressible monotonically increasing sequence of integers. This is the by-product of our sampling algorithm where we first linearize the 3D data arrays and select the points in sequence. We take advantage of the existing lossless compression techniques already integrated in VTK [49] and ParaView [50] for their XML file formats. It has been suggested that fast lossless compression integrated into the *in situ* pipelines can yield time savings for the user with minimal compute time [51]. Testing available compression schemes we empirically observed the well-balanced lzma [52] compressor offers space savings of greater than 74% for the Isabel dataset, as shown in Table 2.

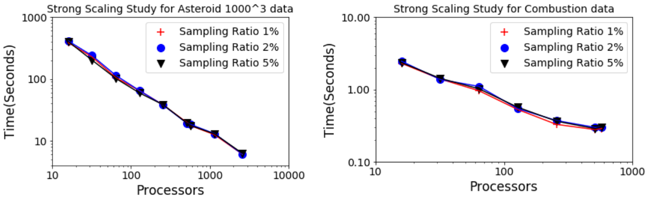
TABLE 3
Quality comparison with ZFP at varying storage limitations (No value indicates ZFP is unable to compress to the required storage size).

Signal-To-Noise Ratio for Isabel dataset				
Region	Sample Rate	Our Method	ZFP (Fixed Rate)	ZFP (Fixed Error)
White (Feature)	0.1%	20.50	-	-
	0.5%	24.87	0.91	17.76
	1%	28.66	21.13	26.38
	3%	34.15	35.33	47.87
	5%	35.89	37.97	53.19
Black (Non-feature)	0.1%	7.78	-	-
	0.5%	13.06	2.10	9.65
	1%	17.57	15.59	18.61
	3%	23.19	28.54	38.02
	5%	25.19	31.42	42.90
Overall	0.1%	11.42	-	-
	0.5%	16.63	1.85	18.37
	1%	20.0	11.87	27.42
	3%	26.25	23.11	49.13
	5%	29.7	25.52	54.15

7.2 Comparison with Lossy Compression

Our proposed sampling method is fundamentally different from the lossy compression techniques available (such as ZFP, SZ, etc). Generally, lossy compression methods are well suited to perform overall data compression/decompression and yield orders-of-magnitude larger compression ratios than lossless compression on floating-point data [53]. Their biggest drawback is that they generally need full data reconstruction for post hoc analysis. In our sampling-based method, we can provide local data reconstruction without needing to reconstruct the full datasets. Users can use our samples as a preview to the data outputs and select the regions for reconstruction and exploration. Also, we have demonstrated using our sampling approach that the sampled data can be used for feature-driven queries and visualization. Additionally, instead of competing with lossy compression methods, the output from our proposed sampling method can be further reduced in size using lossy compression techniques in the $in\ situ$ pipeline (future work). A detailed comparison of our sampling scheme with lossy compression approaches is thus out of scope for this paper. Since our method is primarily geared towards feature preservation at very low sampling rates, in this section we provide a quantitative discussion of the effects of our sampling scheme and a popular lossy compression method ZFP [54] on the feature and non-feature regions of the data while data reduction is performed.

For comparison with the ZFP lossy compression method, we selected the Isabel dataset and compression ratios varying from around 0.1% to around 5%. For ZFP, we used both the fixed-rate mode (where the compressed size can be pre-decided given the storage limitation) and fixed-error mode (where the error can be bounded but the output size cannot be pre-determined). Given our $in\ situ$ scenario, where the storage bandwidth is fixed and predetermined and the data is distributed across processors, the fixed-rate mode is likely to be used. Even though fixed-error mode performs better overall, to make use of it in our scenario we needed to try out multiple error bounds (-a parameter) and select the ones that match closely with the different storage limitations. Even with automation [55], using the fixed-error mode requires multiple compression trials and may never reach both the desired compression ratio given a fixed-error bound. For comparison with our method, we have taken into consideration both the sampled field-value and their index size (compressed with lzma, as discussed in Section 7.1) while computing the sampled storage size. We selected two regions from the Isabel dataset, and the results (in terms of SNR) are presented in Figure 19 (for 1% sampling rate) and Table 3. As can be seen, at low sampling rates, from around 1% and below, our method performs considerably better in the feature region. Specifically, our method outperforms the fixed-rate ZFP in the feature regions, the non-feature regions, and overall. For the extreme case of data reduction (for sampling rate of 0.1%), ZFP could not produce results at that required storage size due to limitations of the ZFP algorithm when requesting for extremely large compression ratios. In these cases, ZFP's required meta-data combined with how ZFP truncates intermediate values results in compression ratios that alternate above and below the target size [55].



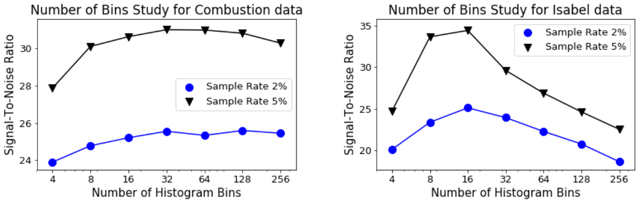
(a) Asteroid data. (b) Combustion data.

Fig. 20. Strong scaling studies for 1%, 2%, and 5% sampling rates for (a) Asteroid and (b) Combustion data.

7.3 Performance

By design, our proposed algorithms are parallelized for distributed memory systems. The two main components of the algorithms are the value-based histogram and gradient-based histogram. Histograms are additive given a common data range and are efficiently parallelized. Computation of gradient is embarrassingly parallel (point-wise operation), and then creating the gradient histogram is similarly efficiently parallelizable. The creation of a joint histogram for our fused method is similarly parallel and scalable. Using these, after the importance function I_F is constructed, then selection of samples using I_F is also embarrassingly parallel. Thus, algorithms are constructed to provide high performance and scalability while running alongside large-scale simulations.

We conducted a strong scaling study on combustion data and an upscaled version of asteroid data (spatial resolution $1000 \times 1000 \times 1000$) using three different sampling ratios: 1%, 2%, and 5%. The experiments were run on a cluster consisting of nodes that contain x86 architectures of many flavors and also Power PC and ARM architectures. For this study, we used homogeneous nodes where the number of processing cores varied from 16 to 2592 for the larger asteroid dataset and 16 to 576 for the smaller combustion dataset. Each node has 36 cores (72 threads) that are Intel Broadwell E5-2695_v4 CPUs with base clock rate 2.10 GHz



(a) Combustion data.

(b) Isabel data.

Fig. 21. Effect of number of bins in the histogram for 2% and 5% sampling rates for (a) Combustion and (b) Isabel data.

and 125 GB of memory. In our experiments, we do not report the data read/write time.

The results are presented in Figure 20 in a log–log scale. For both the asteroid (20(a)) and combustion (Figure 20(b)) datasets, it shows good scaling. From this experiment, it is also observed that the sampling ratios do not have a significant impact on the run-time since we are not considering the data write time in these charts.

7.4 Parameter Analysis

In our sampling scheme, there are primarily two parameters to be tuned by the users.

Sampling ratio: One explicit parameter is the sampling ratio (or sampling rate). This parameter is primarily driven by the storage constraint and can be readily computed as the ratio of desired output size and total data size for each time step.

Number of bins: One implicit parameter is the number of bins to be used for the importance histogram creation. The number of bins should not be too low because that might put rare important scalars into the same bin as less-important scalars, and the important scalars might be lost in the sampling process. On the flip side, if the number of bins is too high, then the histogram will be too finely resolved and the algorithm will become more time consuming. Finely resolved histograms can be too noisy, while coarsely resolved histograms can be biased. Both cases can make it more difficult to capture the basic shape of the distribution. Our study shows that the values between 16 and 64 can generally be safely used as number of bins (e.g., Combustion data in Figure 21(a)) and they produce similar sampled data outputs. We set the number of bins to 32 by default, and this yields satisfactory results in both quality and speed (although Isabel data shows the best performance at 16 bins due to its value distribution, as in Figure 21(b)). Automatic detection of the optimal value of this parameter is a challenge since it is highly data dependent.

In other parameters, as mentioned earlier, users can optionally turn on the error flag to choose the type of error analysis (e.g., max error, SNR, etc.) and provide a block size for that error computation.

7.5 Limitations

In our experiments with varying simulation outputs, the proposed sampling method performs quite well for both sample visualizations and reconstruction quality. Still, our method is a generic data reduction method and does not guarantee all data features can always be preserved. If the features of the data are not in the regions of rare data values or high gradient regions, our proposed method can select less useful samples. Although this can be treated as a limitation of our method for sample visualization, the reconstruction quality should still be high, as demonstrated in our experiments. It is also possible that rare values in the data set may be noise and not of interest. For example, simulations that first need to run for a few time steps before producing features, can produce time steps where there are not many feature regions and rare values are not interesting. For example, cosmology simulations such as Nyx [56] are often initialized with random Gaussian fields and there are no features to be visualized until the simulation has run for a while. In this scenario, the samples from the initial time-steps may not be visually interesting. Finally, this method is currently applicable for regular grid data sets. For non-regular grid data sets, the algorithm will need to be modified such that correct histograms and gradients are computed and voxel volumes are taken into account while generating the samples. This is part of our future work.

8 Conclusion and Future Work

In this work, we presented a novel sampling algorithm that preserves the important regions of a scientific dataset as point samples. We used the data distribution as an indicator for the importance of the scalar values. Using this, we assigned higher importance to the rare values where it is more likely to be the feature or region of importance. After the identification of more important scalars, we incorporated the local smoothness as the second indicator for importance. We created a 2-dimensional acceptance histogram that determines the importance of a sample point based on its value and local gradient. With this data-driven sampling scheme, we studied the different hybrid sampling methods that successfully satisfy the user queries. We demonstrate the capabilities of query-driven visualization with our representative samples. We used multiple real-world large-scale datasets to show the usefulness of our proposed system.

In the future, we would like to extend our algorithms to handle vector and tensor data products and include other data properties for data saliency computation. We would like to further extend our algorithms to exploit more redundancy by considering coherence across time and across multiple variables as well. For our proposed value-based sampling, we will also explore the opportunities to employ different target distributions apart from our currently used uniform. For effective *in situ* use, aside from our current MPI-based approach, we would also like to use GPUs for further speed up. We also will extend our proposed methods to work with unstructured and point-based datasets.

Acknowledgments

The authors would like to thank the Department of Energy and Los Alamos National Laboratory for the funding and support in carrying out this research. This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration and by the Laboratory Directed Research and

Development program of Los Alamos National Laboratory under project number 20200065DR. (LA-UR-20-20029)

References

[1] C. D. Correa and K. Ma, "Visibility histograms and visibility-driven transfer functions," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 2, pp. 192–204, Feb 2011.

[2] S. Dutta, H. Shen, and J. Chen, "In situ prediction driven feature analysis in jet engine simulations," in *2018 IEEE Pacific Visualization Symposium (PacificVis)*, April 2018, pp. 66–75.

[3] 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*, Nov 2014, pp. 43–50.

[4] 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*, vol. 23, no. 1, pp. 811–820, Jan 2017.

[5] S. Lakshminarasimhan, N. Shah, S. Ethier, S. Klasky, R. Latham, R. Ross, and N. F. Samatova, "Compressing the incompressible with isabela: In-situ reduction of spatio-temporal data," in *Euro-Par 2011 Parallel Processing*, E. Jeannot, R. Namyst, and J. Roman, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 366–379.

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

[7] A. Biswas, S. Dutta, J. Pulido, and J. Ahrens, "In situ data-driven adaptive sampling for large-scale simulation data summarization," in *Proceedings of the Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization*, ser. ISAV '18. New York, NY, USA: ACM, 2018, pp. 13–18. [Online]. Available: http://doi.acm.org/10.1145/3281464.3281467

[8] A. Biswas, S. Dutta, H. Shen, and J. Woodring, "An information-aware framework for exploring multivariate data sets," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2683–2692, Dec 2013.

[9] S. Bruckner and T. Möller, "Isosurface similarity maps," in *Proceedings of the 12th Eurographics / IEEE - VGTC Conference on Visualization*, ser. EuroVis'10, Chichester, UK, 2010, pp. 773–782. [Online]. Available: http://dx.doi.org/10.1111/j.1467-8659.2009.01689.x

[10] I. Viola, A. Kanitsar, and M. E. Groller, "Importance-driven feature enhancement in volume visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 4, pp. 408–418, July 2005.

[11] ——, "Importance-driven volume rendering," in *IEEE Visualization 2004*, 2004, pp. 139–145.

[12] Y. Peng, L. Chen, and J. Yong, "Importance-driven isosurface decimation for visualization of large simulation data based on opencl," *Computing in Science Engineering*, vol. 16, no. 1, pp. 24–32, Jan 2014.

[13] L. Gosink, J. Anderson, W. Bethel, and K. Joy, "Variable interactions in query-driven visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1400–1407, Nov 2007.

[14] K. Stockinger, J. Shalf, K. Wu, and E. W. Bethel, "Query-driven visualization of large data sets," in *VIS 05. IEEE Visualization, 2005.*, Oct 2005, pp. 167–174.

[15] 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.

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

[17] Y. Su, G. Agrawal, J. Woodring, K. Myers, J. Wendelberger, and J. Ahrens, "Taming massive distributed datasets: Data sampling using bitmap indices," in *Proceedings of the 22Nd International Symposium on High-performance Parallel and Distributed Computing*, ser. HPDC '13. New York, NY, USA: ACM, 2013, pp. 13–24. [Online]. Available: http://doi.acm.org/10.1145/2462902.2462906

[18] Y. Park, M. J. Cafarella, and B. Mozafari, "Visualization-aware sampling for very large databases," *CoRR*, vol. abs/1510.03921, 2015. [Online]. Available: http://arxiv.org/abs/1510.03921

[19] T. T. Nguyen and I. Song, "Centrality clustering-based sampling for big data visualization," in *2016 International Joint Conference on Neural Networks (IJCNN)*, July 2016, pp. 1911–1917.

[20] X.-H. Chen, A. P. Dempster, and J. S. Liu, "Weighted finite population sampling to maximize entropy," *Biometrika*, vol. 81, no. 3, pp. 457–469, 1994. [Online]. Available: http://www.jstor.org/stable/2337119

[21] C.-W. Ko, J. Lee, and M. Queyranne, "An exact algorithm for maximum entropy sampling," *Operations Research*, vol. 43, no. 4, pp. 684–691, 1995. [Online]. Available: https://doi.org/10.1287/opre.43.4.684

[22] M. C. Shewry and H. P. Wynn, "Maximum entropy sampling," *Journal of Applied Statistics*, vol. 14, no. 2, pp. 165–170, 1987. [Online]. Available: https://doi.org/10.1080/02664768700000020

[23] T. Rapp, C. Peters, and C. Dachsbacher, "Void-and-cluster sampling of large scattered data and trajectories," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2019.

[24] 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)*, 2011pages = 89-96, doi = 10.1109/LDAV.2011.6092322,.

[25] 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. [Online]. Available: http://dx.doi.org/10.2312/EGPGV/EGPGV11/101-109

[26] 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*, ser. CLADE '08. ACM, 2008, pp. 15–24. [Online]. Available: http://doi.acm.org/10.1145/1383529.1383533

[27] 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)*, 2011, pp. 9–14.

[28] 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: ACM, 2017, pp. 42–46. [Online]. Available: http://doi.acm.org/10.1145/3144769.3144778

[29] S. Dutta, J. Woodring, H. W. Shen, J. P. Chen, and J. Ahrens, "Homogeneity guided probabilistic data summaries for analysis and visualization of large-scale data sets," in *2017 IEEE Pacific Visualization Symposium (PacificVis)*, April 2017, pp. 111–120.

[30] K. Wang, N. Shareef, and H. Shen, "Image and distribution based volume rendering for large data sets," in *2018 IEEE Pacific Visualization Symposium (PacificVis)*, April 2018, pp. 26–35.

[31] Y. C. Ye, T. Neuroth, F. Sauer, K. 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)*, Oct 2016, pp. 65–74.

[32] 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.

[33] A. Tikhonova, C. D. Correa, and K. Ma, "Explorable images for visualizing volume data," in *2010 IEEE Pacific Visualization Symposium (PacificVis)*, March 2010, pp. 177–184.

[34] ——, "Visualization by proxy: A novel framework for deferred interaction with volume data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1551–1559, Nov 2010.

[35] S. Li, N. Marsaglia, C. Garth, J. Woodring, J. Clyne, and H. Childs, "Data reduction techniques for simulation, visualization and data analysis," *Computer Graphics Forum*, vol. 37, no. 6, pp. 422–447, 2018. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13336

[36] H. Carr, J. Snoeyink, and M. van de Panne, "Flexible isosurfaces: Simplifying and displaying scalar topology using the contour tree," *Computational Geometry*, vol. 43, no. 1, pp. 42 – 58, 2010, special Issue on the 14th Annual Fall Workshop.

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

[38] J. Albert, *Bayesian computation with R*. Springer Science & Business Media, 2009.

[39] E. Lawrence, S. V. Wiel, and R. Bent, "Model bank state estimation for power grids using importance sampling," *Technometrics*, vol. 55, no. 4, pp. 426–435, 2013.

[40] 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*, vol. 34, no. 11, pp. 2274–2282, Nov 2012.

[41] G. Kindlmann and J. W. Durkin, "Semi-automatic generation of transfer functions for direct volume rendering," in *IEEE Symposium on Volume Visualization (Cat. No.989EX300)*, 1998, pp. 79–86.

[42] T. Zhang, Z. Yi, J. Zheng, D. C. Liu, W.-M. Pang, Q. Wang, and J. Qin, "A clustering-based automatic transfer function design for volume visualization," *Mathematical Problems in Engineering*, vol. 2016, pp. 1–13, 2016.

[43] H. Akiba, K. Ma, J. H. Chen, and E. R. Hawkes, "Visualizing multivariate volume data from turbulent combustion simulations," *Computing in Science Engineering*, vol. 9, no. 2, pp. 76–83, March 2007.

[44] L. J. Gosink, C. Garth, J. C. Anderson, E. W. Bethel, and K. I. Joy, "An application of multivariate statistical analysis for query-driven visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 3, pp. 264–275, March 2011.

[45] J. Patchett and G. Gisler, "Deep water impact ensemble data set," *Los Alamos National Laboratory, LA-UR-17-21595, available at http://dssdata. org*, 2017.

[46] G. R. Gisler, T. Heberling, C. S. Plesko, and R. P. Weaver, "Three-dimensional simulations of oblique asteroid impacts into water," *Journal of Space Safety Engineering*, vol. 5, no. 2, pp. 106–114, 2018.

[47] J. Patchett, F. Samsel, K. Tsai, G. R. Gisler, D. H. Rogers, G. D. Abram, and T. L. Turton, "Visualization and analysis of threats from asteroid ocean impacts."

[48] D. Tao, S. Di, H. Guo, Z. Chen, and F. Cappello, "Z-checker: A framework for assessing lossy compression of scientific data," *The International Journal of High Performance Computing Applications*, vol. 33, no. 2, pp. 285–303, 2019. [Online]. Available: https://doi.org/10.1177/1094342017737147

[49] W. J. Schroeder, K. M. Martin, and W. E. Lorensen, "The design and implementation of an object-oriented toolkit for 3d graphics and visualization," in *Proceedings of Seventh Annual IEEE Visualization '96*, Oct 1996, pp. 93–100.

[50] U. Ayachit, *The paraview guide: a parallel visualization application*. Kitware, Inc., 2015.

[51] J. Patchett and J. Ahrens, "Optimizing scientist time through in situ visualization and analysis," *IEEE Computer Graphics and Applications*, vol. 38, no. 1, pp. 119–127, Jan 2018.

[52] I. Pavlov, "Lzma sdk (software development kit)," January 1999. [Online]. Available: https://www.7-zip.org/sdk.html

[53] S. W. Son, Z. Chen, W. Hendrix, A. Agrawal, W. keng Liao, and A. Choudhary, "Data compression for the exascale computing era - survey," *Supercomputing frontiers and innovations*, vol. 1, no. 2, 2014. [Online]. Available: http://superfri.org/superfri/article/view/13

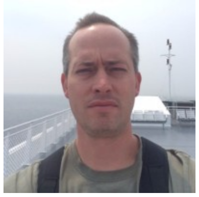
[54] P. Lindstrom, "Fixed-rate compressed floating-point arrays," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2674–2683, Dec 2014.

[55] R. Underwood, S. Di, J. C. Calhoun, and F. Cappello, "FRaZ: A generic high-fidelity fixed-ratio lossy compression framework for scientific floating-point data," in *To appear in the 2020 IEEE International Parallel and Distributed Processing Symposium, IPDPS 2020, New Orleans, LA, USA, May 18-22, 2020*, 2020.

[56] A. S. Almgren, J. B. Bell, M. J. Lijewski, Z. Lukic, and E. Van Andel, "Nyx: A massively parallel amr code for computational cosmology," *Astrophysical Journal*, vol. 765, no. 1, 3 2013.

Ayan Biswas is a scientist in the Data Science at Scale team (CCS-7) at Los Alamos National Laboratory. His research interests include exascale data analysis and reduction, in situ workflows, uncertainty quantification, statistical analysis and high-dimensional data visualization. He also has vast experience in working with vector fields and information theory applications for visualization and analysis. He received his PhD in Data Visualization from The Ohio State University in 2016. Contact him at ayan@lanl.gov.

Soumya Dutta is a staff scientist in Data Science at Scale team at Los Alamos National Laboratory. He received his B.Tech. degree in Electronics and Communication Engineering from West Bengal University of Technology in August 2009, and M.S. and the Ph.D. degree in Computer Science and Engineering from the Ohio State University in May 2017 and May 2018 respectively. His research interests are statistical data summarization and analysis; in situ data analysis, reduction, and feature exploration; uncertainty analysis; and time-varying, multivariate data exploration. Contact him at sdutta@lanl.gov.



Earl Lawrence is a statistician at Los Alamos National Laboratory. He is an expert in uncertainty quantification and the application of statistics to problems in physics and engineering. He earned his high school diploma in 1996 from Bad Axe High School in Michigan's Thumb. Since then, he has learned a great many things that are both useful and interesting. He thinks that all of the Star Wars movies are good in their own way and that most of the people who complain about the newer ones are overlooking obvious flaws in the older ones.

John Patchett received a BA in 1995 and has worked at the Los Alamos National Laboratory (LANL) since 1997. He received an MS in CS in 2011 from the University of New Mexico and a PhD in 2017 from TU Kaiserslautern. Currently John is the Deputy Team Leader for LANL's Data Science at Scale Team and is the Production Visualization Project Lead for LANL's Advanced Simulation and Computing (ASC) Program. He is interested in Large-scale, parallel distributed memory, scientific data visualization and analysis for developing effective workflows for simulation data interpretation and understanding.

Jon C. Calhoun is an Assistant Professor in the Holcombe Department of Electrical and Computer Engineering at Clemson University. He received a B.S. in Computer Science from Arkansas State University in 2012, a B.S. in Mathematics from Arkansas State University in 2012, and a Ph.D. in Computer Science from the University of Illinois at Urbana-Champaign in 2017. His research interests lie in fault tolerance and resilience for high-performance computing (HPC) systems and applications, lossy and lossless data compression, scalable numerical algorithms, power-aware computing, and approximate computing.

James Ahrens of Los Alamos National Laboratory (LANL) is the founder and design lead of ParaView, a widely adopted visualization and data analysis package for large-scale scientific simulation data. Dr. Ahrens graduated in 1996 with a Ph.D. in computer science from the University of Washington. Following his graduate studies, he joined LANL as a technical staff member. At Los Alamos, he is the leader of an awesome data analysis and visualization team of twenty staff, postdocs and students, as well as a national leader of programmatic initiatives important to the Department of Energy's (DOE) National Nuclear Security Administration (NNSA) Advanced Simulation and Computing (ASC) program and the Office of Science (SC) Advanced Scientific Computing Research (ASCR) programs.