Sampling for Scientific Data Analysis and Reduction

Ayan Biswas, Soumya Dutta, Terece L. Turton, James Ahrens

Abstract With exascale supercomputers on the horizon, data-driven in situ data reduction is a very important topic that potentially enables post hoc data visualization, reconstruction, and exploration with the goal of minimal information loss. Sophisticated sampling methods provide a fast approximation to the data that can be used as a preview to the simulation output without the need for full data reconstruction. More detailed analysis can then be performed by reconstructing the sampled data set as necessary. Other data reduction methods such as compression techniques can still be used with the sampled outputs to achieve further data reduction. Sampling can be achieved in the spatial domain (which data locations are to be stored?) and/or temporal domain (which time steps to be stored?). Given a spatial location, data-driven sampling approaches take into account its local properties (such as scalar value, local smoothness etc.) and multivariate association among scalar values to determine the importance of a location. For temporal sampling, changes in the local and global properties across time steps are taken into account as importance criteria. In this chapter, spatial sampling approaches are discussed for univariate and multivariate data sets and their use for effective in situ data reduction is demontrated.

Ayan Biswas
Los Alamos National Lab, Los Alamos, NM, USA, e-mail: ayan@lanl.gov

Soumya Dutta
Los Alamos National Lab, Los Alamos, NM, USA, e-mail: sdutta@lanl.gov

Terece L. Turton
Los Alamos National Lab, Los Alamos, NM, USA, e-mail: tlturton@lanl.gov

James Ahrens
Los Alamos National Lab, Los Alamos, NM, USA, e-mail: ahrens@lanl.gov

1 Introduction

Conceptually, sampling can be loosely defined as the selection of a subset from a collection. In the context of a large scale simulation code, sampling to decrease the output data size is reasonably common. Choosing a regular data output scheme of writing every nth time step is a ubiquitous approach to sampling. As mentioned in the motivating chapter of this book, when very large-scale scientific simulations are running on increasingly powerful supercomputers, one can only store a very small fraction of the generated data. Data movement and I/O restrictions will require more time between data dumps and naive time step selection as a data reduction technique risks losing important events that may occur between regularly scheduled output dumps.

Sampling techniques select m objects from the original pool of M elements with $m \leq M$ for an effective data reduction method. For data reduction and sampling for exascale codes, machine compute capability is orders of magnitude higher than I/O rates, and this translates to $m << M$, i.e., m is orders of magnitude smaller than M. Thus, for in situ large-scale data reduction purposes, simple sampling approaches are no longer sufficient to address the data reduction needs. Adaptive and data-driven sampling techniques become necessary to ensure that the data saved to disk has the relevant statistical properties of the original data and that the information saved prioritizes rare and/or important features and events in the data.

Sampling for data reduction is an attractive approach for in situ data reduction for various reasons. Sampling allows the users to select a representative subset of the raw data points where the true location and the data values for the sampled points are preserved. Downsampled data statistically represents the original data, preserving features and relationships in the data. Use of samples in the post hoc exploration phase does not always require expensive data reconstruction as the reduced data set can be visualized/queried to get an overview of the data.

With these benefits in mind, sampling becomes an attractive in situ solution. Similar to the other in situ data reduction methods discussed in this section of the book, the main goal of sampling is preservation of important data properties or *features* given the limited I/O bandwidth and storage capabilities. In this chapter, scalar and multivariate data sets are addressed. The concept of *importance* is introduced with a discussion of how to define it in an automated data-driven approach to demonstrate how sampling can be a useful tool for preserving the *important* regions. Section 2 briefly discusses the foundations related to in situ sampling and visualization. Section 3 describes different scalar sampling methods for scientific data sets. Section 4 focuses on the sampling methods for multivariate data sets. In situ performance is discussed in Section 5, with limitations of the sampling methods covered in Section 6. Future directions for in situ data-driven sampling in included in Section 7.

2 Prior Work

The development and deployment of novel sampling methods for in situ data reduction and feature preservation is built upon work across multiple disciplines. A flexible many-core capable infrastructure is critical for modern HPC simulation codes to access in situ analysis such as sampling. Bauer et al. [9] provide an overview of in situ infrastructures available to simulation codes for deployment of analysis routines. As HPC heads towards exascale, there are several commonly used infrastructures for scientific applications that can be leveraged to enable in situ sampling. These include ParaView/Catalyst [1, 6], VisIt/LibSim [14], Ascent [30], SENSEI [7], and ADIOS [35].

Each of these in situ APIs provides the visualization and analysis support necessary for in situ sampling use cases and details about many of these infrastructures can be found elsewhere within this book.

Sampling is one of many possible data reduction techniques available to address concurrency challenges at exascale. Son et al. provide a survey of data compression techniques [41] for exascale computing. Lossless compressors such as BLOSC [5] and FPZIP [33] are attractive to avoid introducing biases into the data. However, lossless compression typically does not provide sufficient data reduction to fully solve I/O bandwidth and storage issues. Modern compressors usually allow both lossless and lossy modes, allowing the user to set parameters such as error bounding. Transform methods, such as ZFP [32], SPECK [27] or wavelet approaches such as Li et al. [31], model the data, retaining the most important coefficients in the modelling scheme. Compressors based on predictive algorithms include SZ [19, 45] and FPZIP.

Another common data reduction technique is to move data analysis, visualization, or feature detection in situ, writing out reduced data extracts rather than full raw data sets. Image-based data abstracts have been shown to provide post hoc data analysis flexibility [2, 46] (see, for example, the chapter on Cinema in this book). Reduction through statistical summarization is another approach [25, 48, 52].

Sampling as a data reduction technique is more similar to non-compression data reduction approaches. That said, sampling techniques can often be used in combination with compression to achieve higher levels of overall data reduction.

Information theory [16, 40, 47] forms a theoretical foundation used across many aspects of analysis and visualization of large scientific data sets (see for example [13]). Drawing from information theory, Nouanesengsy et al. [37] developed ADR, Analysis-Driven Refinement, in which user-defined importance metrics are used to select a sparse data set for fast post hoc analysis and visualization. Entropy maximization was used by Biswas et al. ([11]) in the development of an adaptive in situ sampling methodology that prioritized rare but important events in the data. Dutta et al. ([20]) developed a multivariate approach for statistical data summarization and downsampling based on pointwise information theoretic measures.

Woodring et al. [51] introduced and demonstrated a stratified random sampling scheme in a cosmology code. Sampling based on bitmap indexing was first used by Su et al. [43]. Wei et al. [50] proposed IGStS, *information guided stratified sampling*, to downsample data sets while preserving important regions in the data. Park

et al. ([38]) advocated for a visualization aware sampling approach that minimized a loss function based on visualization goals. Sampling in the context of visualization also led Battle et al. ([8]) to develop ScalaR which performs data reduction on the fly for database queries returning results too large for effective interactive visualization. Likewise, with *sampleAction*, Fisher et al. ([22]) focused on improving the visualization experience by visualizing incremental results.

As described in this chapter, the data-driven sampling techniques draw heavily on importance metrics to develop sampling algorithms that prioritize the data of interest to the domain scientist. Readers who may need approaches in addition to or instead of sampling should also consider exploring the other data reduction chapters included in this book.

3 Sampling using Scalar Data Importance

3.1 Motivation for Generic Scalar Sampling

Scalar fields are commonly found across scientific domains and simulations. In the simplest case, a 2D or 3D spatial domain is discretized into smaller regular sized regions and a scalar quantity e.g., *temperature* T, is computed at each discrete location. Although irregular shaped regions are also possible, in this chapter will focus on the regular grid scenario. After the scalar field *temperature* is produced by the simulation, domain scientists use that data to understand the features and phenomenon of interest, e.g., temperature can be used to explore and map the critical global currents that drive global warming and climate change.

A common workflow is for the domain scientist to analyze the features of interest in the data and develop data reduction methods that are specific to the scalar field and features of interest. When the scientist needs to perform a similar feature-driven analysis of a different variable, for example, *pressure* P, the analysis workflow may need to incorporate a different algorithm tuned to this new variable of interest. From the perspective of optimizing the time and effort of the domain scientist, this can be an inefficient workflow.

An alternative approach is to use more generic scalar field sampling methods. This approach is particularly useful in the in situ scenario as the same algorithm can potentially be used across multiple scalar fields during a run, significantly reducing overall I/O and storage needs. An overview of two generic scalar field sampling approaches is first presented before moving on to more sophisticated data-driven sampling methods.

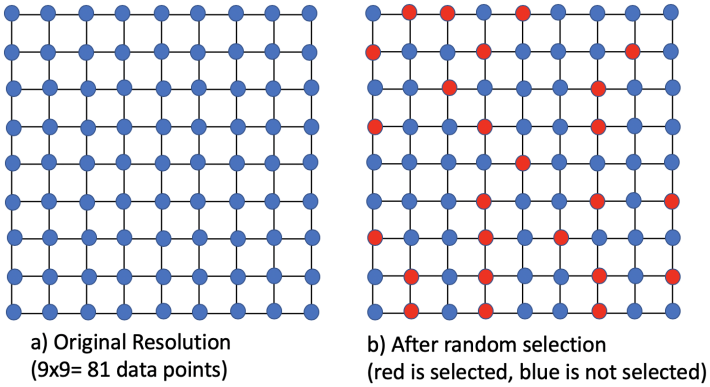


Fig. 1 Schematic example showing random sampling for regular grid data. a) original data b) after randomly selecting 25 out of 81 data points.

3.2 Methods for Scalar Field Sampling

3.2.1 Random Sampling

Random sampling methods are popular due to their simplicity and have been heavily used for various sampling purposes. In a simple random sampling method, each data value has equal probability of being selected. For stratified random sampling, the data is first divided into strata (groups) and then random sampling is applied from within each strata. A schematic example of random sampling can be seen in Figure 1.

Neither simple nor stratified random sampling take into account domain science knowledge or scientific goals. Likewise, neither methods assigns priority to any specific data values that might be of particular importance to the scientist. Hence, although these methods preserve the overall data distribution, important features of the data are not explicitly retained and some or all of these features may be unintentionally lost in the sampling process.

For a regular grid data set, random sampling will produce a particle data set as a result of the sampling. If N_{rand} samples are to be generated, then 4-tuples $\{x, y, z, val\}$ for each location will need to be stored where x, y, z represent the locations of the particles and v is some variable specific to the simulation. This amounts to $4N$ storage.

3.2.2 Regular Sampling

Another well known and commonly used sampling method is naive regular sampling. This method sub-samples the data by regularly selecting data points using a predefined interval. A schematic example is shown in Figure 2. Similar to the random sampling, this method also preserves the overall data distribution and yields similar

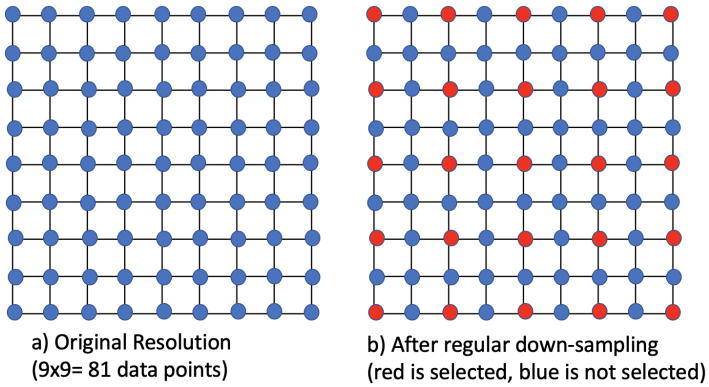


Fig. 2 Schematic example showing regular sampling for regular grid data. a) original data b) after regular selection of 25 out of 81 data points.

statistics as the original data. However, again, there is no notion of feature preservation in the data generated by the sub-sampling scheme. After sampling, regular sampling keeps the regular grid structure. Thus the sampled data output is still on a regular grid and requires N_{reg} storage when N_{reg} samples are to be stored as only the values are stored at each grid point.

3.2.3 Sophisticated Data-driven Sampling Approach

Sampling approaches that are based on random and regular selection have a limitation when used for scientific data analysis and reduction. Although the samples generated from such methods provide a good approximation to the original data distribution, all the data points are given equal *importance* when making the selection. This is not always the ideal way to choose a subset of values when dealing with data from scientific simulations. Generally, such simulations contain *features of interest* that are more important to the domain experts compared to the other parts of the data. Another key point is that these features are potentially much less probable in the data when compared to the chance of occurrence for non-feature regions. Based on these ideas, an in situ algorithm can be devised that can still be generic (i.e., applicable to a diverse set of simulations and scalar fields), but that prioritizes the important feature regions of the data while sampling.

Information theory provides methods applicable to developing data-driven sampling approaches that preferentially save data of interest to the domain scientist. Using the guiding principle that rare values in a data set are more likely to be a feature and the more abundant data values are likely to be background or non-feature regions, a data-driven sampling method can be formulated by assigning more importance to a data value that has lower probability of occurrence, and assigning lower importance to the samples that are more abundant. This simple procedure will likely

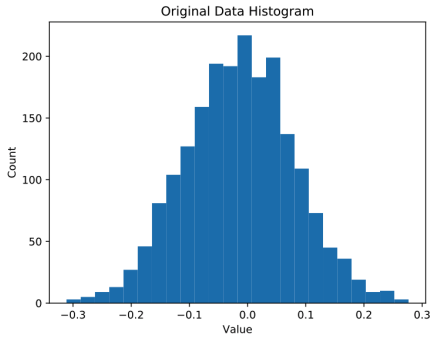
retain the important regions of the data. There are many different ways this can be implemented.

Rather than introducing different user-defined parameters to the system for in situ use, the concept of Shannon's Entropy from information theory can be used to formulate a fast and effective algorithm. Shannon's entropy, H, provides the total information content [17] of a random variable X. This is given as: $H(X) = -\sum_{x \in X} P(x) \log_2 P(x)$, where $P(x)$ is the probability of x, where $x \in X$. Now the principle of maximum entropy can be used for creating generic sampling methods.

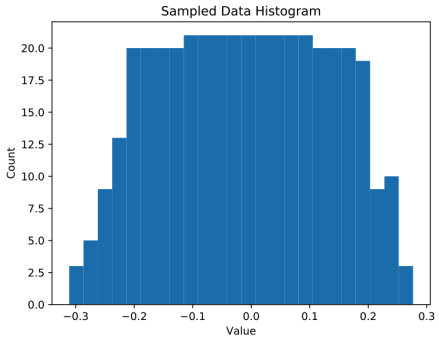
The principle of maximum entropy states that the maximum entropy state of a random variable is the best representation when no other information is available [29]. Using this for a generic sampling method without knowing the features of the data, the best samples would be the ones that maximize the output entropy. Since a uniform distribution has maximum entropy, the goal of maximum entropy sampling will be to try to select equal number of samples for each scalar value irrespective of its abundance in the data set. Ideally, if selecting C samples for all the scalars, then the scalars that have higher occurrence in the data will automatically get a lower probability of acceptance. Similarly, the rare values will have higher chance of selection.

Figure 3 illustrates this concept. From a Gaussian distribution with the mean, $\mu = 0.0$, and the variance, $\sigma^2 = 0.01$, 2000 random data points are taken. The data histogram is shown in Figure 3(a). Computing entropy for this data set using 24 bins results in 3.9 bits. Selecting 20% samples from this data using the entropy-maximizing sampling method as discussed before, results in a histogram similar to Figure 3(b) with an entropy of 4.45 bits. This method of sampling maximizes the entropy from these samples. An *acceptance histogram* can now be defined as a histogram that plots the data values along the x-axis while the y-axis denotes the probability of getting selected for that scalar. This can be seen in Figure 3(c). From this figure, it is easy to observe that the values with higher frequency in the original data were given lower importance and vice versa. An algorithmic representation is given in Algorithm 1.

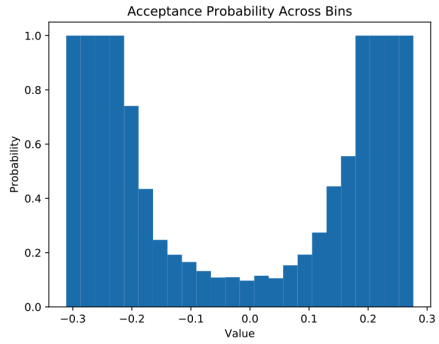
An application of this data-driven intelligent sampling method to the Hurricane Isabel data set [49] is shown in Figure 4. For this data set, the feature of interest is the hurricane eye as shown in Figure 4(a). For demonstration purposes, the Pressure field from time step 25 is chosen. The effect of applying the three different scalar sampling methods (random, regular, and data-driven entropy-maximizing sampling) on this data can be seen. The sampling rates used were 0.5% for random and entropy-based data-driven intelligent sampling method (since they store the point locations) and 1.5% for regular sampling (as it does not need to store the locations). All the sampled data outputs are reconstructed back to original size for comparison purposes. As can be seen from Figure 4, the data-driven entropy-maximizing method (as shown in Figure 4c)) outperforms the random (Figure 4d)) and regular (Figure 4e)) sampling methods in retaining a more complete visual representation of the hurricane eye.



(a) Original data histogram



(b) After entropy maximization



(c) Acceptance histogram

Fig. 3 Illustration of information-driven sampling method via entropy maximization. a) histogram of original data points, b) histogram of the sampled data where the entropy of resulting histogram is maximized c) Acceptance histogram showing the probability of the values of each bin getting accepted after sampling. (This image is reprinted from our previous work [11])

3.3 Sample Analysis and Reconstruction

Often the scientists want to visualize the full-resolution data for exploring important features interactively. Using sub-sampled data requires a technique that can recon-

Input: Sim generated data at each time step
Output: Acceptance histogram
n_k = samples to be taken from full data
N = total points from full data
nbins = number of bins
$n_{samps} = n_k \div nbins$: samples to be taken from each bin
remaining-samples = N
H = CreateHistogram(Data,nbins)
count,bin-edges = SortBinsAccordingToTheirCount(H)
while *not all bins are visited* **do**
  i=0;
  if *count[i]* < n_{samps} **then**
    samples-taken = count[i];
  else
    samples-taken = n_{samps};
  end
  P_i = samples-taken ÷ count[i]
  remaining-samples = remaining-samples - samples-taken
  remaining-bins = nbins - i
  n_{samps} = remaining-samples ÷ remaining-bins
  i=i+1;
end
Use P_is as the probabilities for the corresponding bins of the acceptance histogram

Algorithm 1: Algorithm for the in situ generation of an acceptance histogram.

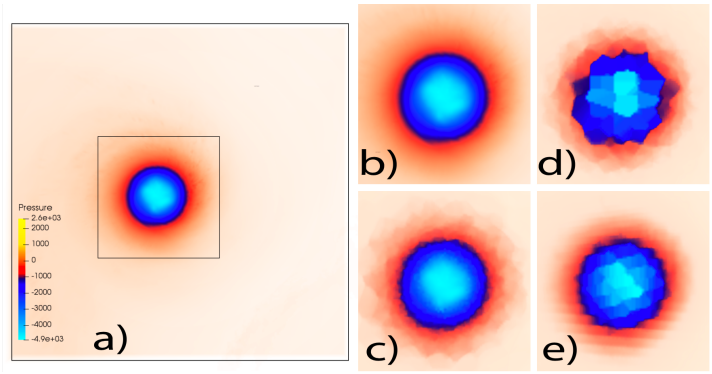


Fig. 4 Sampling results from Hurricane Isabel data set. a) original data b) zoomed in view of the feature (hurricane eye) region. c) reconstruction using data-driven sampling method (sampling ratio 0.5%). d) reconstruction using random sampling method (sampling ratio 0.5%). e) reconstruction using regular sampling method (sampling ratio 1.5%). (This image is reprinted from our previous work [11]).

struct the full-resolution data from the sampled data points. The reconstruction of the full-resolution data can be performed using a nearest-neighbor or linear interpolation based technique. Nearest-neighbor interpolation is faster but generally less accurate. Given a location where a value is to be assigned in the reconstruction phase, this method assigns the value of the nearest sample location. Linear interpolation method

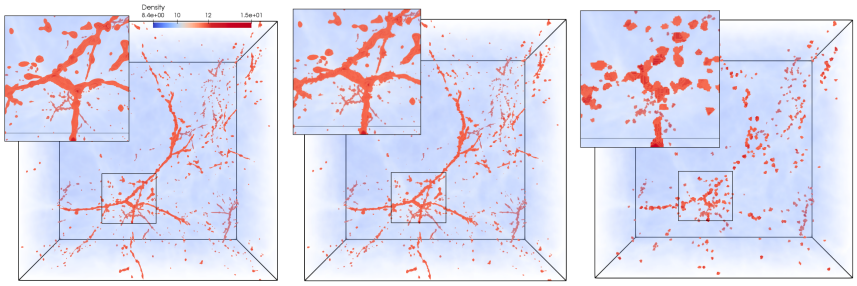


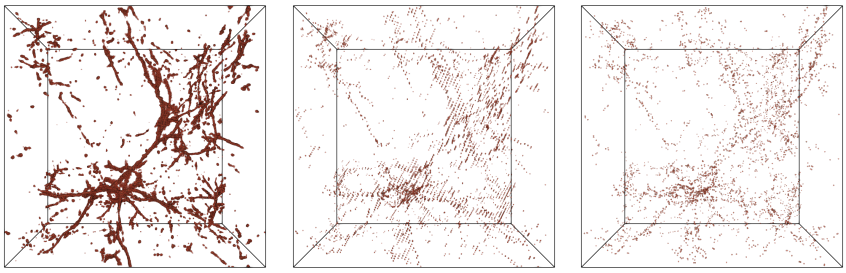
Fig. 5 Volume visualization of sampling results (sampling ratio 1%) using the Nyx simulation. a) Original data and zoomed into a feature region with a dark matter halo. b) Reconstruction using data-driven intelligent sampling method. c) Reconstruction using stratified random sampling method. (This image is reprinted from our previous work [11])

is more time-consuming but often more accurate. In this method, first a 3D convex hull is created using the sampled points and then a polygonal mesh is created using Delaunay triangulation. Then, for each grid point in the reconstruction grid, the simplices in the Delaunay mesh are used and the value is linearly interpolated from the simplex vertices that encloses the current grid point. Once the data is reconstructed (using nearest-neighbor or linear), all analysis/visualization techniques (including the ray casting-based volume rendering that is shown here) can be used to explore the full-resolution data.

3.4 In situ Analysis and Quality Comparison

To demonstrate the performance of the three sampling methods, the Nyx cosmology simulation code [4] is used. The Nyx data resolution was $512 \times 512 \times 512$. The Nyx simulation produces a regular grid data set with multiple variables. Out of these, the baryon density is one of the most important ones to analyze since it provides the particle concentration information tracing back to the origin of universe. In this density field, high density regions correspond to dark matter *halos* forming over time in the universe. These high density regions are therefore the most important to the scientist. In Figure 5, visualizations of the reconstructed samples are shown for comparison purposes at 1% sampling rate for random and data-driven method. Figure 6 visualizes the samples coming out of the sampling methods with even lower sampling ration: 0.5% sampling rate for random and data-driven method and 1.5% for regular method. From both of these figures, it can be observed that the data-driven method preserves the features of the data much better compared to the regular and random methods.

For a quantitative comparison of the quality across these methods, Pearson's correlation coefficient is calculated for both the Hurricane Isabel and Nyx data



(a) (b) (c)

Fig. 6 Point rendering results from Nyx simulation. a) using data-driven intelligent sampling method (sampling ratio 0.5%). b) using regular sampling method (sampling ratio 1.5%). c) using stratified random sampling method (sampling ratio 0.5%). (This image is reprinted from our previous work [11])

Table 1 Quantitative similarity comparison of different sampling method produced images to the image produced by the original data under same rendering configuration.

	Data-driven Samp	Random Samp	Regular Samp
Hurricane Isabel data	0.978	0.921	0.961
Nyx cosmology data	0.987	0.865	0.881

sets. This calculation is performed using volume rendered images from the original data and from the three sampling methods for the same camera angle and transfer function. Then using red, blue and green as three channels, the Pearson correlation value for each sampling method is calculated against the original data. This result is presented in Table 1. As can be seen from this table, the data-driven intelligent sampling method clearly outperforms the other two methods.

Thus, it can be seen that intelligent scalar sampling methods can provide high quality and light-weight data reduction capabilities for scalar fields in large scale data sets. The next section discusses the case where a simulation produces multiple variables to see how information theory can motivate effective multivariate sampling approaches.

4 Sampling using Multivariate Association

4.1 Motivation for Multivariate Sampling

Large-scale simulations commonly produce data sets containing multiple variables. The above section described sampling techniques that work on a single variable at a time while sub-sampling the data. However, data analysis applications may require analysis of multiple variables together. Hence, similar to the univariate sampling technique, multivariate data sampling techniques are also important. Multivariate

sampling can help in reducing a set of simulation variables together. For multivariate data sets, sub-sampled data must preserve the relationship and correlations between variables so that when a post hoc analysis is conducted using the sub-sampled data, the multivariate data features will be the same as an analysis done on the full data set.

Several previous studies have shown that the relationship among multiple variables can be complicated [12, 28, 34]. To summarize such variables together, first, one needs to identify the relationship among them. An effective way of performing sub-sampling of multivariate data would be to sample the points with higher fidelity from the region where the variables show strong statistical association among them. Previous studies have shown that such statistically associated regions often contain multivariate features that are of interest to the application experts for detailed analysis. For example, in a hurricane simulation, the hurricane eye is an important feature, which can be characterized by a spatial region with low pressure and high-velocity values [24, 25]. Similarly, to understand the complex turbulent mixing in a combustion simulation, the study of high valued hydroxyl regions together with the stoichiometric mixture fraction variable reveals more information than studying them individually [3]. Therefore, a multivariate data sub-sampling technique that preserves the statistical association among variables will be able to analyze and visualize such features with high accuracy in the post hoc analysis phase. The following sections describe a multivariate data sampling algorithm that uses statistical data associations and multivariate distributions to sub-sample several data variables together and demonstrates the usefulness of the technique by providing various multivariate applications with qualitative and quantitative studies.

4.2 Multivariate Statistical Association-Driven Sampling

One of the primary requirements of a multivariate sampling technique algorithm is to preserve the multivariate properties, i.e, the interdependence among the chosen variables, and their correlation properties so that the sub-sampled data set can be used effectively for multivariate feature analysis. To achieve this, the multivariate sampling algorithm selects samples densely from the regions where the variable combinations show a strong statistical association or co-occurrence. Higher co-occurrence indicates a stronger association among data values. The strength of the association is quantified for each spatial data point considering their value combination from multiple variables. After the quantification of statistical association is done, sub-sampling is performed according to the strength of the association values and data points which show stronger statistical association have a higher chance of getting selected. At the end of the sampling process, an unstructured data set is produced and this reduced data can be stored onto disk for post hoc analysis. Note that, based on the storage budget, the application user can determine the percentage of data points that will be stored for post hoc analysis and thus determine the amount of data reduction that will be achieved through the sampling.

4.2.1 Estimation of Multivariate Pointwise Information

Multivariate data sampling first requires a criterion to quantify the importance of each data point in the multivariate sense. Each data point in the multivariate data has a value tuple consisting of values from each variable. Consider two variables, X and Y. At each data point, there is a value pair (x, y) where x is a specific value of variable X and y a value for Y. Next, a measure is needed that can quantify the importance of each such value pair so as to select data points that are more informative than others in order to perform sub-sampling. Using information theory, the importance of such value pairs can be estimated using an information-theoretic measure called `pointwise mutual information` (PMI). Pointwise mutual information was first introduced in the works of Church and Hanks [15] for the quantification of the word association. PMI measures the strength of the statistical association of each value pair, thus for each data point. For two random variables X and Y, then the PMI value for the value pair (x, y) can be formally defined as:

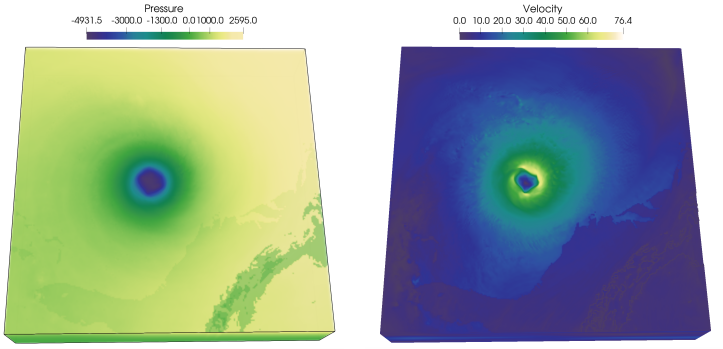
$$PMI(x, y) = \log \frac{p(x, y)}{p(x)p(y)} \tag{1}$$

Here, $p(x)$ is the probability of a particular occurrence x of X, $p(y)$ is the probability of y of variable Y and, $p(x, y)$ is their joint probability. When $p(x, y) > p(x)p(y)$, $PMI(x, y) > 0$, which means x and y have higher statistical association between them. When $p(x, y) < p(x)p(y)$, then $PMI(x, y) < 0$. This condition indicates that the two observations follow a complementary distribution. Finally, when $p(x, y) \approx p(x)p(y)$, then $PMI(x, y) \approx 0$ refers to the case where the variables are statistically independent. It is important to note that the mutual information (MI) $I(X;Y)$ is the expected PMI value over all possible instances of variables X and Y [18].

$$I(X;Y) = E_{(X,Y)}[PMI(x, y)] \tag{2}$$

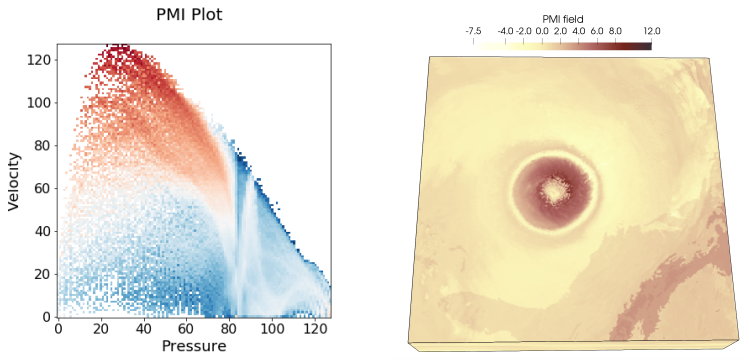
Given this information-theoretic measure, if a value pair has higher PMI value, then it is more likely that the data point associated with it will be selected in the sub-sampled data set. Previous work has shown that regions with high PMI values often correspond to the multivariate features in the data set [21], and hence, selecting data points using their PMI values will also ensure the preservation of important multivariate features in the data set.

To demonstrate the concept of PMI,first consider two variables: Pressure and Velocity from the Hurricane Isabel data set. Figure 7 shows the volume visualization of these two variables. To estimate the PMI values for each data point for these two variables, one first estimates their probability distributions in the form of normalized histograms. Then for each value pair in the 2D histogram bin, the PMI value is estimated using Equation 1. A 2D PMI plot is shown in Figure 8a. The X-axis of the plot shows values of Pressure and the Y-axis shows values of Velocity. Since the computation of PMI values were done using a histogram, the axes in plot 8(a) show the bin IDs. The corresponding scalar value for each bin ID can be easily estimated from the range of data values for each variable. In this plot, the white regions in



(a) Pressure field visualization. (b) Velocity field visualization.

Fig. 7 Visualization of Pressure and Velocity field of Hurricane Isabel data set. The hurricane eye at the center of Pressure field and the high velocity region around the hurricane eye can be observed. (This image is reprinted from our previous work [20].)



(a) PMI plot of Pressure and Velocity field. (b) PMI volume of Pressure and Velocity field.

Fig. 8 PMI computed from Pressure and Velocity field of Hurricane Isabel data set is visualized. Figure 8(a) shows the 2D plot of PMI values for all value pairs of Pressure and Velocity, Figure 8(b) provides the PMI field for analyzing the PMI values in the spatial domain. It can be seen that around the hurricane eye, the eye-wall is highlighted as high PMI-valued region which indicates a joint feature in the data set involving Pressure and Velocity field. (This image is reprinted from our previous work [20].)

the plot represent value pairs with zero PMI values and red regions indicate high PMI values. It can be observed that the low Pressure and moderate to high Velocity values have high PMI values and therefore higher statistical association. Using the PMI values for each data point, a new scalar field can be constructed, namely the PMI field where each data point reflects the pointwise statistical association. In Figure 8b, the PMI field for the Pressure and Velocity variable is shown. It can be seen that the darker brown region has higher values of PMI and is highlighting the eye-wall

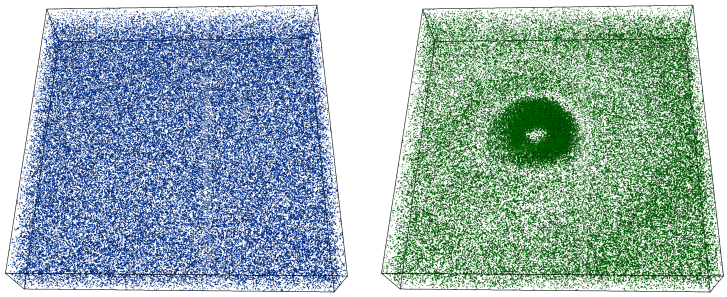
of the hurricane, an important multivariate feature in the data set. Hence, sampling using PMI values will select more data points from the eye-wall region since the data points there have higher PMI values and by doing so, the eye-wall feature will be preserved with higher detail in the sub-sampled data set.

Generalized Pointwise Information. The pointwise mutual information measure allows us to analyze two variables at a time. To estimate the PMI values for more than two variables, a generalized information theoretic measure can be used, called specific correlation [18]. Specific correlation can be formally defined as:

$$SI(x_1, x_2, .., x_n) = log \frac{p(x_1, x_2, ..., x_n)}{p(x_1)p(x_2)...p(x_n)} \tag{3}$$

where $p(x_i)$ represents the probability of an observation x_i for the i^{th} variable X_i, and $p(x_1, x_2, ..., x_n)$ refers to the joint probability of the value tuple $(x_1, x_2, ..., x_n)$. Note that, specific correlation is a generalized extension of the pointwise mutual information and can be used to quantify association for data points when more than two variables are used.

The pointwise information measures presented above depend on joint distributions of variables, raising the question of how these high-dimensional distributions can be computed effectively. Joint probability distributions can be computed using various modeling approaches, such as parametric distributions, non-parametric distributions, etc. Among parametric distribution models, Gaussian mixture model (GMM) is well known for its compactness as only model parameters are necessary to be stored during the calculation. However, the estimation of parameters for high-dimensional Gaussian mixture models using an Expectation-Maximization (EM) technique [10] can be computationally expensive. In contrast, non-parametric models such as Kernel Density Estimation (KDE), Histograms are other alternatives that can be used to estimate joint probability distributions. The computation of high-dimensional KDE is expensive since a significant number of kernel evaluations are required. Furthermore, the storage increases as dimensionality increases. In contrast, the computation of histogram-based distributions is comparatively faster, but the standard high-dimensional histogram representations are not storage efficient. Note that, for a large number of variables, in general, all the above approaches suffer from curse of dimensionality. However, sparse histogram representations reduce storage footprint significantly. Recently, to address the issues of dimensionality, a new technique for high-dimensional histogram estimation has been proposed [36]. This technique is storage efficient and can be computed efficiently in a distributed parallel environment. Another effective and alternative way of modeling high-dimensional distributions is the use of statistical Copula functions [26]. A Copula-based distribution representation only stores the individual independent distributions and the correlation information among the modeled variables. From this Copula model, the joint probability can be queried. This approach reduces the computational cost and storage cost significantly while estimating high-dimensional distributions as shown in [26]. It is also important to note that, in practice, multivariate features are mostly defined using two to three variables in combination, and hence sparse histogram can



(a) Visualization with randomly sampled points.

(b) Proposed statistical association driven sampled points.

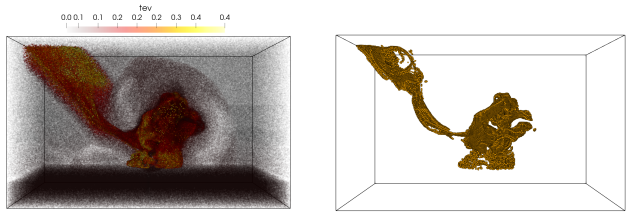
Fig. 9 Sampling result in Hurricane Isabel data set when Pressure and Velocity variables are used. Figure 9(a) shows results of random sampling and Figure 9(b) shows results of the proposed pointwise information-driven sampling results for sampling fraction 0.03. By observing the PMI field presented in Figure 9(b), it can be seen that the proposed sampling method samples densely from the regions where the statistical association between Pressure and Velocity is stronger (Figure 9(b)). (This image is reprinted from our previous work [20].)

be used to estimate the joint distribution for PMI calculation. If higher-dimensional distributions are required, techniques proposed in [36], [26] can be used.

4.2.2 Pointwise Information-driven Multivariate Sampling

This section covers the multivariate sampling process that uses the aforementioned pointwise information measures as the multivariate sampling criterion. The sampling method accepts data points with a higher likelihood if they have higher values of pointwise information. Therefore, regions with higher PMI values (such as the hurricane eye-wall region as seen in Figure 8(b) will be sampled densely compared to regions with relatively low PMI values in the final sub-sampled data set.

The multivariate sampling method accepts a user-specified sampling fraction (α, where $0 < \alpha < 1$) as an input parameter and produces a sub-sampled data set with $n = \alpha \times N (n < N)$ data points where N represents the total number of data points. First, a joint histogram is constructed using all the variables that will be used for sampling. Note that, for a bi-variate case, this results in a 2D histogram. Each histogram bin in this joint histogram represents a value pair and a PMI value can be estimated for each histogram bin. Therefore, the normalized PMI values corresponding to each histogram bin can be used as a sampling fraction for that bin and the bins with higher PMI values will be contributing more to the sample selection process. For example, if a histogram bin has a normalized PMI value of 0.8, then 80% of the data points from this bin will be selected. Note that selecting sample points in this way ensures that the higher PMI valued data points are prioritized in the sampling process and by doing so, data points with a stronger statistical association



(a) All the data points sampled by the proposed sampling method.

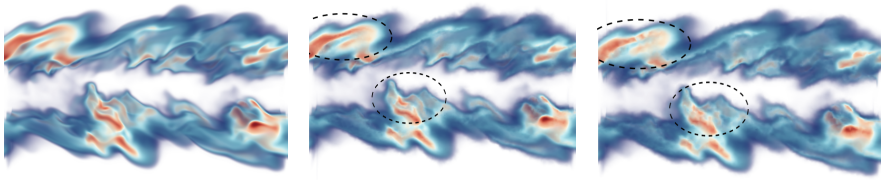
(b) Result of the query when applied on raw data.

(c) Result of the query when applied on PMI-based sampled data.

(d) Result of the query applied on randomly sampled data.

Fig. 10 Visualization of multivariate query driven analysis performed on the sampled data using Asteroid impact data set. The multivariate query 0.13 < tev < 0.5 AND 0.45 < v02 < 1.0 is applied on the sampled data sets. Figure 10a shows all the points selected by the proposed sampling algorithm by using tev and v02 variable. Figure 10b shows the data points selected by the query when applied to raw data. Figure 10c shows the points selected when the query is performed on the sub-sampled data produced by the proposed sampling scheme and Figure 10d presents the result of the query when applied to a randomly sampled data set. The sampling fraction used in this experiment is 0.07. (This image is reprinted from our previous work [20].)

are preserved in the sub-sampled data. The interested reader can find more details of this sampling method in [20]. An example of this multivariate sampling is shown in Figure 9. Figure 9(b) shows the sub-sampled data points for the Hurricane Isabel data set when Pressure and Velocity fields are used. It can be observed that data points from the eye-wall region were selected densely since the data points in that region have high PMI values. In Figure 9(a), a randomly sub-sampled field is shown. By comparing Figure 9(b) with Figure 9(a), it is evident that the proposed sampling method preserves the statistically associated regions in the sub-sampled data set and accurate post hoc multivariate feature analysis using such a data set can be done.



(a) Visualization using raw data.

(b) Reconstructed data visualization using proposed sampling algorithm.

(c) Reconstructed data visualization using random sampling algorithm.

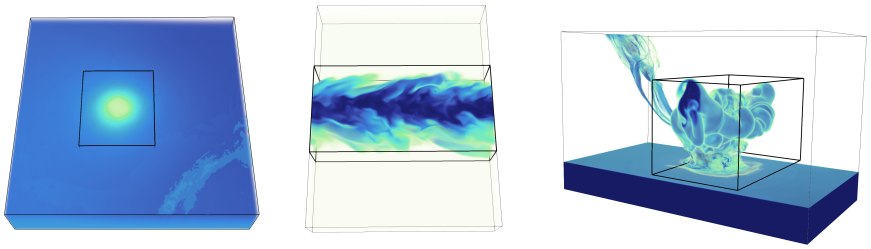
Fig. 11 Reconstruction-based visualization of Y_OH field for a turbulent combustion data set. Linear interpolation is used to reconstruct the data from the sub-sampled data sets. Figure 11(a) shows the result of the original raw data. Figure 11(b) provides the reconstruction result from the sub-sampled data generated by the proposed method. Figure 11(c) presents the result of reconstruction from randomly sampled data. The sampling fraction used in this experiment is 0.05. (This image is reprinted from our previous work [20].)

4.3 Applications of Multivariate Sampling

4.3.1 Sample-Based Multivariate Query-Driven Visual Analysis

Query-driven visualization (QDV) is a popular technique for analyzing multivariate features in a scientific data set [23, 42, 25]. Query-driven analysis helps the expert to focus quickly on the region of interest, effectively reducing their workload. In this example, the sub-sampled data set is used directly for answering domain specific queries using an asteroid impact data set. The Deep Water Asteroid Impact data set [39] was generated at the Los Alamos National Laboratory to study Asteroid Generated Tsunami (AGT). The data set contains multiple variables. The volume fraction of water variable, denoted by v02, and the temperature variable denoted by tev, are used in this study. First the data set is sub-sampled using the above association-driven multivariate sampling algorithm, retaining 7% of the data points. The interaction between tev and v02 can be studied by performing the following multivariate query: $0.13 < tev < 0.5 \;\; AND \;\; 0.45 < v02 < 1.0$.

The result of the query-driven analysis is shown in Figure 10. Figure 10(a) shows all the 7% sample points that were selected by the sampling method. In Figure 10(b) the result of the query is shown when applied to the ground truth data. Figure 10(c) depicts the query results when the sub-sampled data generated from multivariate PMI-based sampling is used. Compared to the result generated from random sampling, Figure 10(d), one can see that the multivariate association-driven sampling can answer the query with higher accuracy.



(a) ROI for Hurricane Isabel data.

(b) ROI for combustion data.

(c) ROI for asteroid data.

Fig. 12 Regions of interest (ROI) of different data sets used for analysis. Figure 12(a) shows the ROI in Isabel data set, where the hurricane eye feature is selected. Figure 12(b) shows the ROI for the Combustion data set, where the turbulent flame region is highlighted. Finally, in Figure 12(c), the ROI for asteroid data set is shown. The ROI selected in this example indicates the region where the asteroid has impacted the ocean surface and the splash of the water is ejected to the environment. (This image is reprinted from our previous work [20].)

4.3.2 Reconstruction-Based Visualization of Sampled Data

The combustion data set provides a second example. The reconstruction technique is as discussed in Section 3.3. An example visualization of a reconstructed data for the Y_OH field of a combustion data set is presented in Figure 11. The sub-sampled data set was generated using mixture fraction and Y_OH fields and in this example, 5% data points were stored. As can be seen, the reconstructed data, Figure 11(b), generated using the data samples produced by the multivariate association-driven sampling technique, produces a more accurate visualization compared to the reconstruction obtained from randomly sampled data set, Figure 11(c). The black dotted regions highlighted in Figure 11(b) and Figure 11(c) show the regions where the reconstruction error is more prominent compared to the ground truth raw data shown in Figure 11(a).

4.3.3 Multivariate Correlation Analysis of the Proposed Sampling Method

While analyzing multivariate data, application experts often study multivariate relationships among variables to explore variable inter-dependencies. Scientific features in multivariate data sets typically demonstrate statistical association in the form of linear or non-linear correlations among variable values. Therefore, it is important to preserve such variable relationships in the sub-sampled data so that flexible post hoc analysis can be done. In this section, the multivariate correlations obtained from the reconstructed data (the reconstruction is described in Section 4.3.2) that used multivariate association driven sampling are compared with the correlation values that were obtained from reconstructed data that used randomly selected samples. For analysis, a feature region (region of interest (ROI)) was selected for each data

Table 2 Evaluation of multivariate correlation for feature regions. The feature regions for each data set are shown in Figure 12 indicated by a black box. (This table is reused from our previous work [20].)

	Raw data		PMI-based Sampling		Random Sampling	
	Pearson's Correlation	Distance Correlation	Pearson's Correlation	Distance Correlation	Pearson's Correlation	Distance Correlation
Isabel Data (Pressure and QVapor)	-0.19803	0.3200	-0.19805	0.3205	-0.1966	0.3213
Combustion Data (mixfrac and Y_OH)	0.01088	0.4012	0.01624	0.04054	0.02123	0.4071
Asteroid Data (tev and v02)	0.2116	0.2938	0.2273	0.2994	0.2382	0.31451

set. The ROI for each data set is shown in Figure 12 using the dark black box. In Table 2 the results are presented. Pearson's correlation coefficient is again used for linear correlation and for measuring non-linear correlation, distance correlation [44] is used. The purpose of this study is to demonstrate that the multivariate association driven sampling technique is able to preserve the correlation among the modeled variables more accurately compared to the random sampling based method. Table 2 demonstrates that when the PMI-based multivariate sampling is used, the correlation in the reconstructed data matches closely with the correlation values obtained from the raw data. The correlation values obtained from random sampling have a higher deviation from the true correlation. This indicates that the multivariate PMI-based sampling technique preserves both the linear and non-linear correlations more accurately compared to the random sampling technique.

5 In situ Performance

An in situ performance study for univariate spatial sampling methods was performed using the Nyx cosmology simulation code [4]. The three sampling methods were implemented in C++ and were integrated into the Nyx simulation code in the `writePlotFile();` in situ I/O routine. As mentioned earlier, the sampling routines were called each time step and the test performed on a cluster with Intel Broadwell E5_2695_v4 CPUs (18 cores per node and 2 threads per core), and 125 GB of memory per node. For the in situ scaling study, the Nyx simulation was run for 100 time steps with $512 \times 512 \times 512$ resolution per time step. For a distributed setting where different parts of the data reside on different nodes, the data-driven univariate sampling algorithm can be efficiently extended to handle such scenarios. Since histograms are additive across data blocks, before starting this method, only MPI-based communications needed are the global data minimum value, maximum value and the local histograms computed based on the global data minimum/maximum. Next, these local histograms can be "reduced" to achieve the global histogram and this algorithm can begin.

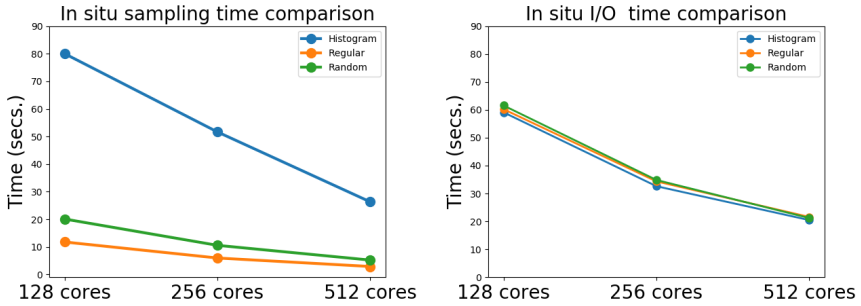


Fig. 13 a) Comparison of in situ sampling performance among the three sampling methods: histogram-based sampling (blue), regular sampling (orange), and random sampling (green). b) Comparison of in situ sampled data I/O times among the three sampling methods: histogram-based sampling (blue), regular sampling (orange), and stratified random sampling (green). (This image is reprinted from our previous work [11])

The in situ results are shown in Figure 13 where the run times for different methods are shown. As can be seen from this figure, all the three methods scale quite well. It is also visible that the entropy-based method is slightly more expensive since it performs more data analysis in situ. However, this increase in time is compensated by the much higher quality samples generated by this method. As shown in Figure 13(b), the in situ I/O time is similar for all the three methods. Table 3 lists the time spent by the sampling methods as a percentage of total simulation time. This table indicates that, although the entropy-based method spends more time than other two naive methods, the percentage time spent compared to the actual simulation time is still negligible (only about 1.5%). Percentage disk I/O time for the sampled data is also shown in this table and as observed, they are quite small fraction of the simulation raw I/O.

Similar performance trends are expected for the multivariate sampling algorithm since it also uses a distribution-based approach. However, the computation of joint probability distributions will require more time compared to the univariate distributions. In this case, to achieve viable in situ computational performance, one can follow the joint probability distribution estimation approach of [36]. Another alternative is to use the Copula-based modeling schemes that are computationally efficient for in situ environments [26].

6 Discussions and Limitations

In this chapter, in situ sampling methods for both univariate and multivariate data sets have been discussed in detail. Examples of data-driven approaches demonstrate their effectiveness compared the naive (albeit faster) regular/random methods. It is

Table 3 In situ percentage timings of different sampling methods and I/O timings w.r.t the simulation timings.

	Hist. Samp %sim time	Reg. Samp %sim time	StRand. Samp %sim time	Hist. I/O %sim I/O	Reg. I/O %sim I/O	StRand. I/O %sim I/O
128 Cores	1.39	0.20	0.35	2.86	2.92	2.97
256 Cores	1.83	0.21	0.37	3.07	3.23	3.28
512 Cores	1.41	0.16	0.28	2.61	2.75	2.70

also important to note that data-driven methods may not necessarily be applicable for all the time steps of a simulation. Specifically, for simulations (such as Nyx) that are initialized on a random field and for first few time steps may have no features in the data. For such cases, it is recommended that users try to employ random/regular sampling methods as the data-driven method may not produce any meaningful samples or capture any interesting features. Due to the lack of features in the data, essentially, data-driven methods will reduce to behaving like random methods. For multivariate sampling, it is expected that the variables used will have some association or correlation among themselves so that the information-theoretic measure PMI will capture such informative data points and the sampling will be guided by it. If the variables are statistically independent, then the univariate sampling technique can be applied instead of the multivariate sampling and as mentioned above, for a collection of independent variables, the multivariate sampling will also behave like a random sampling technique.

7 Future Directions and Conclusion

Looking forward in this research area, there are other in situ data reduction approaches that can leverage interesting sampling methods. Incorporating more data properties when selecting important samples is one such avenue of research. Likewise, these methods can be applied to time-varying aspect of the data as well as considering vector fields. The arrival of exascale architectures present opportunities to incorporate more computationally expensive techniques as the compute resources prioritize in situ approaches over post hoc analysis.

This chapter covered different sampling approaches that can be performed in situ for data reduction. When considering univariate data for spatial sampling, data distribution can be used to identify and save important scalars that are more likely to be features in the data set. Similarly, for multivariate data set, point-wise mutual information can be leveraged to identify locations that capture multivariate importance. Using these light-weight yet scalable methods, data-driven sampling can yield results that are more meaningful to the scientist, compared to naive methods such as random or regular sampling.

Acknowledgement

We would like to thank our Data Science at Scale Team colleagues: D. H. Rogers, L.-T. Lo, J. Patchett, our colleague from the Statistical Group CCS-6: Earl Lawrence, our industry partners at Kitware and other collaborators: C. Harrison, M. Larsen. 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. This research used resources provided by the Los Alamos National Laboratory Institutional Computing Program, which is supported by the U.S. Department of Energy National Nuclear Security Administration under Contract No. 89233218CNA000001. This research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725. The Hurricane Isabel data set has kindly been provided by Wei Wang, Cindy Bruyere, Bill Kuo, and others at NCAR. Tim Scheitlin at NCAR converted the data into the Brick-of-Float format. The Turbulent Combustion data set is made available by Dr. Jacqueline Chen at Sandia National Laboratories through US Department of Energy's SciDAC Institute for Ultrascale Visualization. This research was released under LA-UR-20-21090.

References

1. Ahrens, J., Geveci, B., Law, C.: Paraview: An end-user tool for large data visualization. The Visualization Handbook **717** (2005)
2. Ahrens, J., Jourdain, S., O'Leary, P., Patchett, J., Rogers, D.H., Petersen, M.: An image-based approach to extreme scale in situ visualization and analysis. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 424–434. IEEE Press (2014)
3. Akiba, H., Ma, K., Chen, J.H., Hawkes, E.R.: Visualizing multivariate volume data from turbulent combustion simulations. Computing in Science Engineering **9**(2), 76–83 (2007). DOI 10.1109/MCSE.2007.42
4. Almgren, A.S., Bell, J.B., Lijewski, M.J., Lukić, Z., Van Andel, E.: Nyx: A Massively Parallel AMR Code for Computational Cosmology. apj **765**, 39 (2013). DOI 10.1088/0004-637X/765/1/39
5. Alted, F.: BLOSC (2009). URL http://blosc.pytables.org/. [online]
6. Ayachit, U., Bauer, A., Geveci, B., O'Leary, P., Moreland, K., Fabian, N., Mauldin, J.: Paraview catalyst: Enabling in situ data analysis and visualization. In: Proceedings of the First Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization, pp. 25–29. ACM (2015)
7. Ayachit, U., Whitlock, B., Wolf, M., Loring, B., Geveci, B., Lonie, D., Bethel, E.W.: The sensei generic in situ interface. In: 2016 Second Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization (ISAV), pp. 40–44 (2016). DOI 10.1109/ISAV.2016.013
8. Battle, L., Stonebraker, M., Chang, R.: Dynamic reduction of query result sets for interactive visualizaton. In: 2013 IEEE International Conference on Big Data, pp. 1–8 (2013). DOI 10.1109/BigData.2013.6691708
9. Bauer, A.C., et al.: *In Situ* Methods, Infrastructures, and Applications on High Performance Computing Platforms, a State-of-the-art (STAR) Report. Computer Graphics Forum, Proceedings of Eurovis 2016 **35**(3) (2016). LBNL-1005709

10. Bilmes, J.: A gentle tutorial on the em algorithm including gaussian mixtures and baum-welch. Tech. rep., International Computer Science Institute (1997)
11. Biswas, A., Dutta, S., Pulido, J., Ahrens, J.: 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, ISAV '18, p. 13–18. Association for Computing Machinery, New York, NY, USA (2018). DOI 10.1145/3281464.3281467
12. Biswas, A., Dutta, S., Shen, H., Woodring, J.: An information-aware framework for exploring multivariate data sets. IEEE Transactions on Visualization and Computer Graphics **19**(12), 2683–2692 (2013). DOI 10.1109/TVCG.2013.133
13. Chen, M., Feixas, M., Viola, I., Bardera A.and Shen, H., Sbert, M.: Information Theory Tools for Visualization. CRC Press, Boca Raton, FL, USA (2006)
14. Childs, H., et al.: VisIt: An End-User Tool For Visualizing and Analyzing Very Large Data. In: High Performance Visualization—Enabling Extreme-Scale Scientific Insight, pp. 357–372. CRC Press/Francis–Taylor Group (2012)
15. Church, K.W., Hanks, P.: Word association norms, mutual information, and lexicography. In: Proceedings of the 27th annual meeting on Association for Computational Linguistics, ACL '89, pp. 76–83. Association for Computational Linguistics, Stroudsburg, PA, USA (1989). DOI 10.3115/981623.981633. URL http://dx.doi.org/10.3115/981623.981633
16. Cover, T., Thomas, J.: Elements of Information Theory, 2 edn. Wiley Series in Telecommunications and Signal Processing. Wiley-Interscience, New York, NY, USA (2006)
17. Cover, T.M., Thomas, J.A.: Elements of Information Theory 2nd Edition (Wiley Series in Telecommunications and Signal Processing). Wiley-Interscience (2006)
18. Van de Cruys, T.: Two multivariate generalizations of pointwise mutual information. In: Proceedings of the Workshop on Distributional Semantics and Compositionality, DiSCo '11, pp. 16–20. Association for Computational Linguistics, Stroudsburg, PA, USA (2011). URL http://dl.acm.org/citation.cfm?id=2043121.2043124
19. Di, S., Cappello, F.: Fast error-bounded lossy HPC data compression with sz. In: 2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS), pp. 730–739 (2016). DOI 10.1109/IPDPS.2016.11
20. Dutta, S., Biswas, A., Ahrens, J.: Multivariate pointwise information-driven data sampling and visualization. Entropy **21**(7), 699 (2019)
21. Dutta, S., Liu, X., Biswas, A., Shen, H.W., Chen, J.P.: Pointwise information guided visual analysis of time-varying multi-fields. In: SIGGRAPH Asia 2017 Symposium on Visualization, SA '17, pp. 17:1–17:8. ACM, New York, NY, USA (2017). DOI 10.1145/3139295.3139298. URL http://doi.acm.org/10.1145/3139295.3139298
22. Fisher, D., Popov, I., Drucker, S., schraefel, m.: Trust me, i'm partially right: Incremental visualization lets analysts explore large datasets faster. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '12, p. 1673–1682. Association for Computing Machinery, New York, NY, USA (2012). DOI 10.1145/2207676.2208294. URL https://doi.org/10.1145/2207676.2208294
23. Gosink, L., Anderson, J., Bethel, W., Joy, K.: Variable interactions in query-driven visualization. IEEE Transactions on Visualization and Computer Graphics **13**(6), 1400–1407 (2007). DOI 10.1109/TVCG.2007.70519
24. Gosink, L.J., Garth, C., Anderson, J.C., Bethel, E.W., Joy, K.I.: An application of multivariate statistical analysis for query-driven visualization. IEEE Transactions on Visualization and Computer Graphics **17**(3), 264–275 (2011). DOI 10.1109/TVCG.2010.80
25. Hazarika, S., Dutta, S., Shen, H., Chen, J.: Codda: A flexible copula-based distribution driven analysis framework for large-scale multivariate data. IEEE Transactions on Visualization and Computer Graphics **25**(1), 1214–1224 (2019). DOI 10.1109/TVCG.2018.2864801
26. Hazarika, S., Dutta, S., Shen, H., Chen, J.: Codda: A flexible copula-based distribution driven analysis framework for large-scale multivariate data. IEEE Transactions on Visualization and Computer Graphics **25**(1), 1214–1224 (2019). DOI 10.1109/TVCG.2018.2864801
27. Islam, A., Pearlman, W.A.: Embedded and efficient low-complexity hierarchical image coder. In: Electronic Imaging'99, pp. 294–305. International Society for Optics and Photonics (1998)

28. Jänicke, H., Wiebel, A., Scheuermann, G., Kollmann, W.: Multifield visualization using local statistical complexity. IEEE Transactions on Visualization and Computer Graphics **13**(6), 1384–1391 (2007). DOI 10.1109/TVCG.2007.70615
29. Jaynes, E.T.: Information theory and statistical mechanics. Phys. Rev. **106**(4), 620–630 (1957). DOI 10.1103/PhysRev.106.620
30. Larsen, M., Ahrens, J., Ayachit, U., Brugger, E., Childs, H., Geveci, B., Harrison, C.: The alpine in situ infrastructure: Ascending from the ashes of strawman. In: Proceedings of the In Situ Infrastructures on Enabling Extreme-Scale Analysis and Visualization, ISAV'17, pp. 42–46. ACM, New York, NY, USA (2017). DOI 10.1145/3144769.3144778
31. Li, S., Marsaglia, N., Chen, V., Sewell, C., Clyne, J., Childs, H.: Achieving portable performance for wavelet compression using data parallel primitives. In: Proceedings of the 17th Eurographics Symposium on Parallel Graphics and Visualization, PGV '17, p. 73–81. Eurographics Association, Goslar, DEU (2017). DOI 10.2312/pgv.20171095. URL https://doi.org/10.2312/pgv.20171095
32. Lindstrom, P.: Fixed-rate compressed floating-point arrays. IEEE Transactions on Visualization and Computer Graphics **20**(12), 2674–2683 (2014)
33. Lindstrom, P., Isenburg, M.: Fast and efficient compression of floating-point data. IEEE Transactions on Visualization and Computer Graphics **12**(5), 1245–1250 (2006)
34. Liu, X., Shen, H.W.: Association analysis for visual exploration of multivariate scientific data sets. IEEE Transactions on Visualization and Computer Graphics **22**(1), 955–964 (2016). DOI 10.1109/TVCG.2015.2467431
35. Lofstead, J.F., Klasky, S., Schwan, K., Podhorszki, N., Jin, C.: Flexible io and integration for scientific codes through the adaptable io system (adios). In: Proceedings of the 6th International Workshop on Challenges of Large Applications in Distributed Environments, CLADE '08, p. 15–24. Association for Computing Machinery, New York, NY, USA (2008). DOI 10.1145/1383529.1383533. URL https://doi.org/10.1145/1383529.1383533
36. Lu, K., Shen, H.W.: A compact multivariate histogram representation for query-driven visualization. In: Proceedings of the 2015 IEEE 5th Symposium on Large Data Analysis and Visualization (LDAV), LDAV '15, pp. 49–56 (2015)
37. Nouanesengsy, B., Woodring, J., Patchett, J., Myers, K., Ahrens, J.: ADR visualization: A generalized framework for ranking large-scale scientific data using analysis-driven refinement. In: 2014 IEEE 4th Symposium on Large Data Analysis and Visualization (LDAV), pp. 43–50 (2014). DOI 10.1109/LDAV.2014.7013203
38. Park, Y., Cafarella, M., Mozafari, B.: Visualization-aware sampling for very large databases. In: 2016 IEEE 32nd International Conference on Data Engineering (ICDE), pp. 755–766 (2016). DOI 10.1109/ICDE.2016.7498287
39. Patchett, J., Gisler, G.: Deep water impact ensemble data set. Los Alamos National Laboratory, LA-UR-17-21595, available at http://dssdata. org (2017)
40. Shannon, C.E.: A mathematical theory of communication. SIGMOBILE Mob. Comput. Commun. Rev. **5**(1), 3–55 (2001). DOI 10.1145/584091.584093
41. Son, S., Chen, Z., Hendrix, W., Agrawal, A., Liao, W., Choudhary, A.: Data compression for the exascale computing era - survey. Supercomput. Front. Innov.: Int. J. **1**(2), 76–88 (2014). DOI 10.14529/jsfi140205. URL https://doi.org/10.14529/jsfi140205
42. Stockinger, K., Shalf, J., Wu, K., Bethel, E.W.: Query-driven visualization of large data sets. In: VIS 05. IEEE Visualization, 2005., pp. 167–174 (2005). DOI 10.1109/VISUAL.2005.1532792
43. Su, Y., Agrawal, G., Woodring, J., Myers, K., Wendelberger, J., Ahrens, J.: Taming massive distributed datasets: Data sampling using bitmap indices. In: Proceedings of the 22nd International Symposium on High-Performance Parallel and Distributed Computing, HPDC '13, p. 13–24. Association for Computing Machinery, New York, NY, USA (2013). DOI 10.1145/2462902.2462906. URL https://doi.org/10.1145/2462902.2462906
44. Székely, G.J., Rizzo, M.L., Bakirov, N.K.: Measuring and testing dependence by correlation of distances. The Annals of Statistics **35**(6), 2769–2794 (2007). URL http://www.jstor.org/stable/25464608
45. Tao, D., Di, S., Chen, Z., Cappello, F.: Significantly improving lossy compression for scientific data sets based on multidimensional prediction and error-controlled quantization. In: 2017

IEEE International Parallel and Distributed Processing Symposium (IPDPS), pp. 1129–1139 (2017). DOI 10.1109/IPDPS.2017.115
46. Tikhonova, A., Correa, C.D., Ma, K.: Explorable images for visualizing volume data. In: 2010 IEEE Pacific Visualization Symposium (PacificVis), pp. 177–184 (2010)
47. Verdu, S.: Fifty years of Shannon theory. IEEE Transactions on Information Theory **44**(6), 2057–2078 (1998). DOI 10.1109/18.720531
48. Wang, K., Kewei Lu, Wei, T., Shareef, N., Shen, H.: Statistical visualization and analysis of large data using a value-based spatial distribution. In: 2017 IEEE Pacific Visualization Symposium (PacificVis), pp. 161–170 (2017)
49. Wang, W., Bruyere, C., Kuo, B., Scheitlin, T.: IEEE visualization 2004 contest data set. http://sciviscontest.ieeevis.org/2004/data.html (2004). NCAR
50. Wei, T., Dutta, S., Shen, H.: Information guided data sampling and recovery using bitmap indexing. In: 2018 IEEE Pacific Visualization Symposium (PacificVis), pp. 56–65 (2018). DOI 10.1109/PacificVis.2018.00016
51. Woodring, J., Ahrens, J., Figg, J., Wendelberger, J., Habib, S., Heitmann, K.: In-situ sampling of a large-scale particle simulation for interactive visualization and analysis. Computer Graphics Forum **30**(3), 1151–1160 (2011). DOI 10.1111/j.1467-8659.2011.01964.x
52. Ye, Y.C., Neuroth, T., Sauer, F., Ma, K., Borghesi, G., Konduri, A., Kolla, H., Chen, J.: 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), pp. 65–74 (2016)