An Application of Multivariate Statistical Analysis for Query-Driven Visualization

Luke J. Gosink, Christoph Garth, John C. Anderson, E. Wes Bethel, and Kenneth I. Joy,

Abstract—Driven by the ability to generate ever-larger, increasingly complex data, there is an urgent need in the scientific community for scalable analysis methods that can rapidly identify salient trends in scientific data. Query-Driven Visualization (QDV) strategies are among the small subset of techniques that can address both large and highly complex datasets. This paper extends the utility of QDV strategies with a statistics-based framework that integrates non-parametric distribution estimation techniques with a new segmentation strategy to visually identify statistically significant trends and features within the solution space of a query. In this framework, query distribution estimates help users to interactively explore their query's solution and visually identify the regions where the *combined* behavior of constrained variables is most important, statistically, to their inquiry. Our new segmentation strategy extends the distribution estimation analysis by visually conveying the *individual* importance of each variable to these regions of high statistical significance. We demonstrate the analysis benefits these two strategies provide and show how they may be used to facilitate the refinement of constraints over variables expressed in a user's query. We apply our method to datasets from two different scientific domains to demonstrate its broad applicability.

Index Terms—Query-Driven Visualization, Multivariate Analysis, Kernel Density Estimation.

1 Introduction

Scientific visualization accelerates the discovery process in contemporary science by transforming abstract data into a visual representation that readily conveys comprehensible, meaningful information to the scientist. In the scientific discovery process, Bergeron [6] identifies three functional modalities for visualization use: visual-discovery, visualization-based analysis, and presentation visualization.

Discovery visualization strategies are employed early in the scientific discovery process during initial stages of investigation. These strategies are designed to help scientists uncover new trends or features in data where the scientist is unsure—due to the complexity of the data or the undefined nature of the event being studied—what trends or anomalies to look for. Such strategies typically demand a great deal of I/O bandwidth and computation to support high levels of user interactivity.

Query-based methods are class of highly effective discovery visualization strategies. One query-based approach, Query-Driven Visualization (QDV), is an extremely powerful strategy that couples scalable database indexing technologies, like FastBit [41], [42], [43] and DP-BIS [18], with visualization techniques to efficiently manage large-scale data, perform rapid data analysis, and support unconstrained, interactive exploration of complex data [13], [15], [16], [37].

The term "Query-Driven Visualization" refers to the strategy of restricting computation and cognitive workloads exclusively to records defined to be "interesting" by the scientist. This strategy is realized through the evaluation of user-specified, multidimensional Boolean range queries—e.g. ($temperature < 1200$) AND ($pressure > 2.4$)—that rapidly filter away large portions of non-pertinent data, and allow smaller, more interesting subsets of data to be efficiently analyzed and visualized. QDV provides a flexible analysis component similar to techniques based on brushing with linked views [4], [23], [40]; both visual-exploratory approaches employ a user-driven process for selecting and highlighting important trends in complex data. Unlike brushing methods, however, QDV is inherently and tightly coupled with database technologies that allow QDV strategies to be readily applied to large-scale data.

Query-based visualization research successfully addresses many important areas such as scalability and performance [14], [37], visualizing multitemporal data [13], [16], applying QDV strategies to adaptively meshed domains [16], addressing uncertainty in multivariate queries [13], and extending query-based strategies to address the challenges of visualizing function field data [1]. Researchers have also successfully combined domain-specific knowledge with QDV, e.g., to study network traffic [5] or combustion flame fronts [15].

Comparatively little work, however, has been performed on the explicit study of query solutions, i.e. the set of records selected by a scientist's multidimensional Boolean range query. This underdeveloped area of research highlights a central problem for query-based strategies like QDV: while solutions to queries can help scientists identify interesting visual features, trends, or anomalies within their data, existing query-based research offers little assistance in helping scientists to better understand these events. Combustion flame fronts, vortices, chemical reaction fronts are all examples of very complex scientific phenomenon. Obtaining greater insight into these events with QDV (e.g., their formation, duration, and evolution) requires understanding how all variables—pressure, temperature, etc—*jointly* interact within the solution space of a query. Further, scientists must be able to analyze and understand individual variable trends within the context of these joint interactions—e.g. among all chemical species within the flame front region, which chemical radicals are most significant to the joint trends driving the combustion process?

The challenge is to extend the strengths of QDV with new analysis methods that can help QDV users better understand the solutions to their queries. Such strategies are essential for scientists to progress from stages of discovery visualization to stages of visual analysis and presentation. This work extends the utility of QDV with a statistical framework that enables scientists to attain greater levels of insight into the features and anomalies identified by QDV applications. With this framework, QDV users can identify:

- the individual significance (e.g. central statistical tendencies and important trends) of each variable to the query's solution in the comparative context of all other variables constrained by the query;
- the salient *joint* trends (i.e. trends based on the behavior and interaction of *all* variables combined) that are characteristically important to understanding the query's solution; and,
- how to adjust individual variable constraints in a query to focus on, or exclude, these newly identified trends in subsequent searches.

The new insight provided by our statistical framework will help to accelerate scientists from stages of visual-discovery into stages of visual analysis and presentation.

In our framework, the "statistical structure" of a query is composed of two statistical measures: a global measure that takes into account all variables constrained by the query, and a segmentation, which is based on the localized statistical contribution of all variables to the query's solution.

The global measure is comprised of the estimated joint distribution of the query's solution space. Exploring this joint distribution allows QDV users to interactively explore their query's solution and visually identify the regions where the *combined* behavior of constrained variables is most important or interesting to their search. To provide further insight into the query's joint distribution, we introduce a new segmentation strategy that extends the distribution estimation analysis by visually conveying the *individual* importance of each variable to these regions of high statistical significance. The global and localized measures, when integrated together, facilitate a means for refining variable constraints expressed in the QDV user's query. This framework addresses a critical need in query-based research by providing a domain-agnostic approach that solidifies a path for discovery visualization users to take in order to begin understanding the events and anomalies they have identified in their data.

The main contributions of this work include the following:

- We introduce a statistics-based framework that extends the utility of QDV strategies by helping users to better understand the solutions to their queries. The core of this framework is based on a strategy that integrates non-parametric distribution estimation techniques with a new segmentation strategy to visually identify statistically important variable trends—both individual variable trends and *joint* trends for groups of variables—within the solution space of a query.
- We show how users can use the information obtained from the query's joint distribution and segmentation to refine the constraints over individual variables in a multivariate query.

We demonstrate these methods across data from different application domains. Furthermore, we perform all statistical data processing on the graphics processing unit (GPU) to facilitate quick response times for the QDV user.

In the next section, we discuss work that is germane to our efforts. In Section 3 we present our statistical processing framework, and approach for creating surfaces and segmentations of query solutions. Our surfacing and segmentation enables meaningful query visualization and analysis, as we show in Section 4. We conclude by addressing important implementation issues, with a discussion of GPU implementation and performance in Section 5.

2 Previous Work

2.1 Distribution Estimation in Image Processing and Computer Vision

In the image processing community, distribution functions and methods for estimating distributions, e.g., kernel density estimates (KDE) [33], [34], are used primarily for image classification—i.e. a Boolean segmentation of the image into regions of interest and regions of non-interest. For example, Zhang and Yang [46] utilize KDE and statistical analysis to detect and isolate moving objects of interest, e.g. pedestrians, cars, etc., from streaming images. Liu et al. [28] also demonstrate a KDE-based probabilistic framework for image classification. They use probability distribution functions based on a hybrid KDE and Gaussian Mixture Model (GMM) to isolate moving objects from movie frames while simultaneously removing artifacts like shadows and obstructing foreground.

Mean shift clustering [9], [12], [45], which is based on the mean shift procedure presented by Fukunaga and Hostetler [11], is a common, non-parametric segmentation technique used in computer vision research to facilitate feature space analysis. In mean shift clustering, the feature space of the image is modeled by its estimated joint distribution (obtained through KDE methods). Comaniciu and Meer [9] show that the dense regions in the feature space of the image directly correspond to local maxima in the image's estimated joint distribution. Segmentation of the image is realized by assigning pixels in feature space to the modality nearest the pixel in the image's estimated joint distribution. There is thus one unique segment in the image for every unique local maxima in the image's distribution estimate.

A fundamental difference between the segmentation generated by mean shift clustering and our segmentation strategy is that mean shift clustering is driven by the gradient of the estimated joint distribution, whereas our segmentation is based on the localized statistical significance each variable plays in *constructing* the joint distribution. Hence, mean shift clustering is a post-process performed *after* the KDE calculation, and our segmentation is obtained *during* the process of calculating the KDE. In Section 3.3 we discuss this difference more thoroughly when we introduce our segmentation procedure.

2.2 Distributions in Visualization

In the visualization community distribution functions, e.g. histograms [22], support and facilitate a wide variety of tasks. For example, the volume reconstruction equation used in splatting utilizes a distribution function to calculate and spread the color contribution a given voxel makes to a pixel region in screen space [38]. Westover [39] shows the model used to construct the splatting distribution function (e.g. Gaussian, bilinear) dramatically influences the quality of the rendered image during the splatting process. Crawfis and Max [10] present a cubic spline function for splatting that supports both accurate rendering from all viewing directions and generates images superior to those rendered with a Gaussian distribution kernel. Mueller et al. [31] use distribution kernels of varying size, where the kernel size is based on a given voxel's distance to the viewing plane, to effectively ameliorate aliasing artifacts in splatting. This type of approach is particularly effective when the volume resolution is higher than the image resolution.

Histograms are also used extensively in volume rendering. Ledergerber et al. [26] utilize a moving least squares method to reconstruct the underlying distribution of a series of data points along a single ray. They use the reconstructed distribution to volume render high-quality images with accurate shading. Kniss et al. [25] combine the underlying distribution functions of scalar data values with data attributes, such as gradient magnitude, to derive 2D and 3D transfer functions. These transfer functions provide a powerful and intuitive way to rapidly isolate important visual features (e.g. surfaces) in multivariate data that are not able to be isolated with simple 1D transfer functions. Finally, Lundstrom et al.[30] combine user domain knowledge, in the form of local histogram criteria, into a certainty-based classification strategy to create transfer functions for direct volume rendering. They apply their strategy on magnetic resonance data and show that their constructed transfer functions clearly detect and separate important tissues of interest, e.g. liver, spleen, kidney, during volume rendering.

Recently, the direct relationship between isosurface complexity, histogram distributions, and geometric statistics (i.e., the area, volume, and gradient magnitude corresponding to the surface of a given isovalue) has been established by Carr et al. [7] and Scheidegger et al. [35]. Scheidegger et al. show that isosurface statistics and histograms converge to the same results. They extend this finding by showing how their techniques can be seen as a way to calculate expectations of random variables on isosurfaces. Bajaj et al. [2] compute geometric statistics of isolines and isosurfaces and display them in an interface to assist users performing discovery visualization tasks. In this work, contour trees are used to provide global structure to the observed statistical measures in order to provide visual cues to the user about interesting and important isovalues in scalar data. This work presents an excellent tool to guide scientists tasked with visual discovery. Unfortunately, as the authors' strategy focuses on scalar field data, the work does not immediately lend itself to the challenges of visualizing multivariate data; for example, the important trends observed between multiple variables in high dimensional data.

Linsen et al. [27] focus on feature space analysis for Smoothed Particle Hydrodynamic (SPH) simulations with a d-dimensional binning strategy to estimate distributions for SPH-based scattered data. The resulting distribution gives rise to a hierarchical distribution-based clustering of feature space. However, in comparison to our approach, their method cannot be used to examine the influence individual variables have in the distribution or the distribution generated by the specific subset contained in a query's solution set.

Query-driven visualization offers a rich setting in which to employ distribution functions for analysis. In the following section we apply a statistics-based framework to generate a joint distribution function for a query's solution. We use this distribution to construct accurate surfaces that indicate where important regions of interest lie within the solution space of the query. We also generate a segmentation, based on the contributing distribution of all variables, in order to isolate and visualize important statistical features of interest from the query. This segmentation can be used to help refine variable constraints in the query in order to further investigate regions of interest.

3 Method

Query-Driven Visualization (QDV) is based upon the evaluation and direct visualization of queries over large scientific data [37]. Queries typically take the form of Boolean range constraints upon individual variables of multivariate data. The "solution" to a query are the regions of a dataset for which the variables satisfy the range constraints.

Consider a function $f : R^3 \to R^d$, and a query comprised of lower and upper limits, a and b respectively, upon the range of f. The solution Q to the query may then be defined as:

$$Q := \{p \in R^3 | a \le x \le b\},$$

where $a, b \in R^d$, and x is the vector of variable values associated with point p, e.g. $f(p) = x = (x_0, \ldots, x_d)$. The solution set Q then corresponds to all the points in the spatial domain for which $a_i \le x_i \le b_i$ for $i \in (0, \ldots, d)$.

Classically, query solution sets have been visualized by using a straightforward depiction of their data constituents (i.e. points or cells). This visualization is carried out by rendering each cell that passes the query as a hexahedral cube [16], [37], or a solid sphere [13]. An example of this type of rendering is shown in Fig. 1. In this image, regions of low pressure and high pressure are observed in a hurricane dataset. These regions correspond to areas where pressure is less than -1500 Pascal (green at center), or above 250 Pascal (blue).

Isosurfaces too can be used to visualize query solution sets by defining the function $h : R^d \to [0,1]$:

$$(h \circ f)(p) = \begin{cases} 0 & \text{if } p \notin Q \\ 1 & \text{if } p \in Q \end{cases}$$

Visualizing the surface $(h \circ f)^{-1}(0.5)$ approximates the boundary containing Q with a single surface and more smoothly than cell or sphere-based renderings.

In our approach, however, by estimating the joint distribution of the $p \in Q$, we are able to provide not only a single,

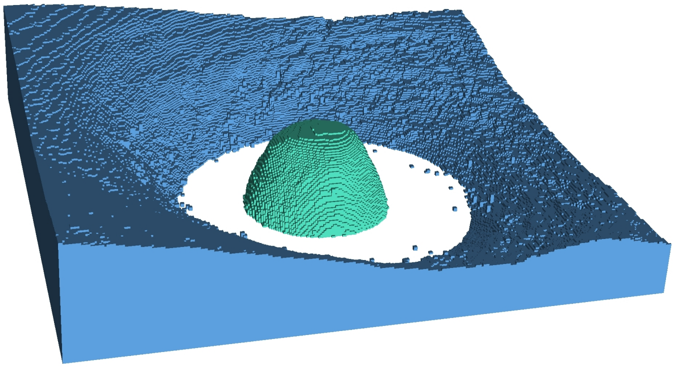


Fig. 1. This figure depicts a typical cell-based rendering used to visualize query solutions in Query-Driven Visualization. In this image, regions of low pressure (green, at center) and high pressure (blue) are visualized for a query that selects regions where pressure is less than -1500 Pascal, or above 250 Pascal.

accurate surface that depicts the query's boundary, but also provide information to the QDV user that enables them to better understand their query's solution.

For example, consider the solution set of a query that constrains multiple variables from a turbulent combustion dataset: e.g. temperature, pressure, as well as chemical species such as methane (CH_4), water (H_2O), and hydroxyl radicals (OH). The isosurface corresponding to the lowest, non-zero estimated density in the query's joint distribution (see the definition of S_{min} in Section 3.2) defines the accurate, smooth boundary for the query's solution. Isosurfaces corresponding to areas of slightly increased density can indicate areas where individual variable distributions are simultaneously in a state of transition. With respect to our combustion example, these regions can indicate where the concentration of chemical species and values for temperature and pressure are all *jointly* changing rapidly within the query region; such areas can be indicative of important events such as flame fronts, extinction regions etc. Isosurfaces corresponding to areas of high density indicate regions where variables possess increased statistical significance. Specifically these regions, and the corresponding range of values associated with each variable in these regions, indicate the locations and range of values for each variable that best characterize, statistically, the query's solution.

The foundation of our method is the computation of the underlying joint variable distribution for multivariate data within a query solution (Section 3.1). Using this joint distribution estimate, we describe the construction of surfaces from the distribution fields (Section 3.2). Finally, we define a segmentation of the query solution based on the localized, statistical significance each variable plays in *constructing* the joint distribution (Section 3.3).

3.1 Distribution Estimation for Queries

We begin our query analysis by constructing a distribution estimation for the multivariate solution space of a query. Kernel Density Estimation (KDE) can be applied to develop a statistical model of the underlying functional behaviour of multiple samples from one or more variables. Consider a set of N observed data samples $x^0, x^1, \ldots, x^N$ from a function $f : R^3 \to R$. The estimation $\hat{f}$ for the underlying distribution of f is:

$$\hat{f}(x) = \frac{1}{Nh}\sum_{i=1}^{N} K\left(\frac{x - x^i}{h}\right), \tag{1}$$

where h is the kernel bandwidth parameter for smoothing, and K is a Gaussian kernel:

$$K(x) = \frac{1}{\sqrt{2\pi}} \exp^{-\frac{x^2}{2}}. \tag{2}$$

To determine the kernel bandwidth parameter, we employ an *adaptive estimate spread* method [36]. This method has been shown to work well for unimodal distributions, while not over-smoothing features in multimodal distributions:

$$h := 0.9\left(min\left(\sigma^2, \frac{R}{1.34}\right)\right) N^{-\frac{1}{5}}, \tag{3}$$

where σ^2 and R are the standard deviation and inner quartile range for the data samples, respectively.

In our work we apply KDE over the solution set of a query. Thus the data samples are the multivariate values $x \in R^d$ corresponding to the points $p \in Q$. Correspondingly, we must construct the joint distribution estimate over multivariate data samples, where the multivariate KDE is defined:

$$\hat{f}(x) = \frac{1}{N\left\{\prod_{j=1}^{d} h_j\right\}} \sum_{i=1}^{N} \prod_{j=1}^{d} K\left(\frac{x_j - x_j^i}{h_j}\right). \tag{4}$$

Here we use unique, per-variable kernel bandwidth parameters h_j computed using (3) to evaluate this multivariate Gaussian kernel.

3.2 Visualizing Queries Using Their Distribution

Previous QDV surfaces have presented a "blocky", binary separation of space due to a lack of interpolation away from points returned by the query engine [37]. In the previous section we defined the construction of distribution estimates for a multivariate query's solution space. Now, we utilize these estimates to define surfaces that bound query regions.

From a joint distribution estimate, a new scalar field $g : R^3 \to R$ is formed, where g maps all elements $p \in Q$ to their distribution values computed in (4). Elements outside the solution set, i.e. $p \notin Q$, are set to zero because they do not contribute to the query's underlying distribution:

$$g(p) = \begin{cases} \hat{f}(f(p)) & \text{if } p \in Q, \\ 0 & \text{otherwise.} \end{cases}$$

We can use the clamped distribution estimate field g to construct surfaces that contain the solution space of the query. To generate a surface that bounds the query solution, we observe that there will be some element $p^{min} \in Q$ with a minimum, non-zero distribution value. The solution to the query may then be visualized by the isosurface $g(p^{min})^{-1}$. We refer to such a surface as the "minimum distribution surface," and denote it simply as S_{min}. Because we map multivariate data samples to a scalar kernel density estimates (KDE), we can visualize query surfaces with common isosurfacing algorithms such as Marching Cubes [29] or raycasting.

Given $g : R^3 \rightarrow R$, it is possible to visualize surfaces corresponding to higher distribution values than S_{min}, with the goal of query analysis and refinement. Surfaces formed from increasingly higher distribution values will contain the regions for which data samples are more representative of the total data selected by the user's query. By examining these surfaces, the user is able to refine their variable constraints intuitively in a visual manner, and without losing the information critical to their query.

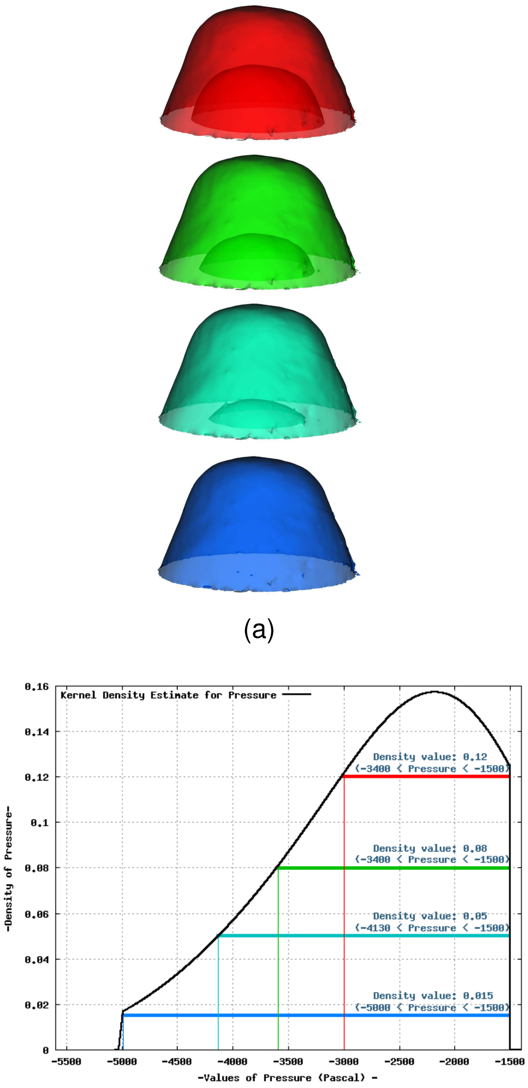
We illustrate this refinement procedure in Fig. 2. In 2b, we see the estimated distribution constructed from elements $p \in Q$ by a query selecting regions of low pressure in a hurricane dataset: pressure ≤ -1500 Pascal. Exploring this distribution with isosurfaces corresponding to increasing distribution values can help the user to locate new visual features and refine the constraints of the original query. We illustrate this constraint refinement in Fig. 2a; we see the surface corresponding to the original query's solution for low pressure in the bottom image (blue). This surface corresponds to the minimum distribution surface S_{min}. Using transparency, we see the effect of examining surfaces for distribution values greater than S_{min}; moving up from the bottom image in Fig. 2a, elements with distribution values greater than 0.05 (blue-green), 0.08 (green), and 0.12 (red). Note that these isosurfaces also correspond to selecting an increasingly smaller subset of points $p \in Q$ with high distribution values. From Fig. 2b we see that these subsets also correspond to an increasingly tighter range of values for pressure. With this type of exploration, the user can visually explore the solution (i.e. distribution) of their query to obtain information regarding the distribution behavior of its variables.

3.3 Multivariate Query Segmentation

When visualizing an estimated joint distribution constructed from (4), localized regions containing high distribution values can be the result of a single variable's contribution, or the cumulative contribution of several variables. To generate deeper insight into the query solution and help the user better understand regions of local maxima and minima in the joint distribution, we employ a strategy of feature analysis through segmentation.

There are many multi-labeled data segmentation algorithms [3], [21], [24], [32], but the most effective and common segmentation employed for the analysis of KDE is non-parametric mean shift clustering [9]. While mean shift clustering can classify and reveal distinct and major modalities in a distribution, it can't generate insight into the important variable trends occurring within these regions. For example, given a specific local maxima, or a group of maxima, in a query's joint distribution, a scientist may be interested in knowing:

- Are all variables constrained by the query well represented in these distribution features, or only certain ones? If certain variables are predominant, which ones and how predominant?
- Conversely, if a variable's distribution is *not* strongly contributing to specific modalities in the estimated joint distribution, where *is* this variable's distribution most predominant in contributing to the query's KDE?



(a)

(b)

Fig. 2. This figure visualizes distribution data calculated from a query that selects regions of low pressure in a hurricane dataset: pressure $\leq$ -1500 Pascal. (a) shows specific isosurfaces corresponding to *decreasing* statistical density in this data: (top to bottom) 0.12, 0.08, 0.05, and 0.015, where the surface rendered at 0.015 is the S_{min} surface for the query. (b) relates these distribution values to an increasingly refined range of values constrained by the query: e.g., the blue surface contains values (-5000 $<$ *pressure* $<$ -1500), and the red surface contains values (-3400 $<$ *pressure* $<$ -1500).

Mean shift clustering can't generate enough insight to answer these questions, it can only identify the regions where the *joint* behavior of variables is significant. Hence, for Query-Driven Visualization (QDV) applications, mean shift clustering can't help the scientist progress from stages of visual discovery into stages of visual analysis and presentation.

To help answer these questions and generate deeper insight into the query's joint distribution, we present a new segmentation strategy based on each variable's individual contribution to the query's KDE. Let $\hat{f}_j$ denote the estimated univariate distribution of the j^{th} variable. We then extract the portion of the query solution Q_j associated with variable j as:

$$Q_j := \left\{ p \in Q \middle| \ \hat{f}_j(x) > \hat{f}_k(x) \quad \forall k \neq j \right\}.$$

Taken together, the subsets $Q_0, \dots, Q_d$ form a partition of the query solution set Q. Note that in computing the joint distribution estimate in (4), the univariate distributions can be obtained by accumulating the individual Gaussian inner product terms for each j—thus the segmentation is obtained efficiently, with minimal additional overhead for computation and storage.

Interpreting Segmented Regions

From a high level the segmented regions visually convey—in the comparative context of all other variables constrained by the query—the individual significance of each variable to the query's solution. Visualizing segments concurrently by using isosurfaces (see Fig. 4 in Section 4.1, and Fig. 7 in Section 4.2) or direct volume rendering shows *where* the distribution of each variable is most important in defining the visual feature, trend, or anomaly the scientist has discovered.

Segmented regions, when visualized concurrently with local maxima regions in the query's KDE (see Fig. 9 in Section 4.2), indicate which variables predominantly contribute to statistically important features in the query's joint distribution. Contrariwise, if a variable distribution is *not* strongly contributing to specific modalities in the estimated joint distribution, segmented regions can also indicate where a variable's distribution *is* most predominant in contributing to the query's KDE. We illustrate this strategy in Section 4.2 with a methane combustion dataset to show that regions corresponding to high distribution values in the query are predominantly influenced by *temperature* and CO_2 behavioral trends, and *not* trends due to *pressure*.

From a low level, the corresponding range of values for the $p \in Q_j$ contain a subset of values for the variable j that are important and significant for the user. To attain greater insight from the segmented regions, it is therefore important to consider the univariate distribution (i.e. histogram) of each variable j as found throughout the query's solution space Q, versus the variable's segmented region Q_j. In our analysis we employ the univariate distribution estimates $\hat{f}_j$ for each segmented region *as it is defined exclusively to* Q_j and visualize the corresponding minimum distribution surfaces to represent each segmented region. As we show in Section 4.1, it is often the case that the univariate distribution obtained for Q_j isolates distribution modalities from Q. This observation can then be used to perform multivariate query refinement. More specifically, it is possible to refine constraints over variable j to focus upon or exclude a modality isolated in Q_j. We illustrate this refinement strategy in Section 4.1 on a hurricane dataset. In this example, we refine an initial query by using a modality isolated in *temperature*'s segmented region.

We now apply our strategy—using KDE, segmentation, and query refinement—to two separate datasets to demonstrate its utility in generating greater insight for Query-Driven Visualization strategies.

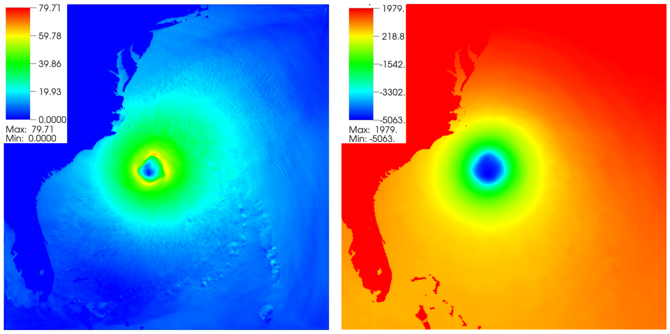


Fig. 3. Slices through the velocity (left) and pressure (right) scalar fields of the hurricane dataset. In Section 4.1 we utilize these variables in a query that selects regions of low velocity and low pressure.

4 Visualization Applications and Analysis

We apply our new method to two datasets and demonstrate our ability to generate surfaces that bind the query's solution and perform distribution-based segmentation. In the first example, this segmentation is utilized to refine the constraints expressed for our query. In the second example, this segmentation is concurrently visualized with regions of high distribution in the query's KDE to identify which variables are predominant in forming the solution to the query.

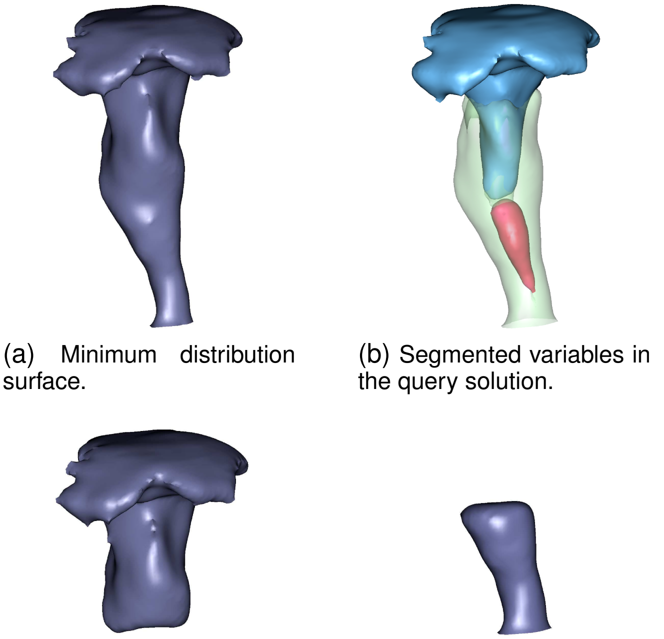
4.1 Hurricane Dataset

This dataset was generated by a simulation modeling a hurricane over a 48 hour period. This dataset consists of 13 variables over a grid size of 300 x 300 x 90, and is composed of 48 timesteps. In this experiment, we evaluate a query that selects all cells, from a single timestep, where records contain both low pressure, low wind velocity and fall in a broad range of temperature: pressure $\leq$ -350 Pascal AND velocity $\leq$ 10 mph AND -70 $\leq$ temperature $\leq$ 20 Celsius. The constraining characteristics of this query roughly approximate the features that classify the hurricane's eye in this dataset. In our analysis, we will analyze the variable-based segmentation of this region, and demonstrate our approach for multivariate query refinement.

We apply our method to the set of points that have been selected by the query after intersecting the regions of low pressure, low velocity, and broad temperature. For illustrative purposes, we see in Fig. 3 slices through the hurricane's velocity (left) and pressure (right) scalar fields. Temperature is not depicted as the query selects all points based on values for temperature.

In Fig. 4a, we see the query solution set Q visualized by the minimum distribution surface S_{min}. Here we render our surfaces using a traditional Marching Cubes implementation over the raw data of the scalar joint distribution field. The surface roughly resembles the center of the hurricane event.

We next visualize the segmentation that we obtained when constructing the joint distribution for this query. In Fig. 4b, we see the results of the segmentation performed on the query's



(a) Minimum distribution surface.

(b) Segmented variables in the query solution.

(c) Refined query where the solution surface shows the upper region of Fig. 4a.

(d) Refined query where the solution surface shows the lower region of Fig. 4a.

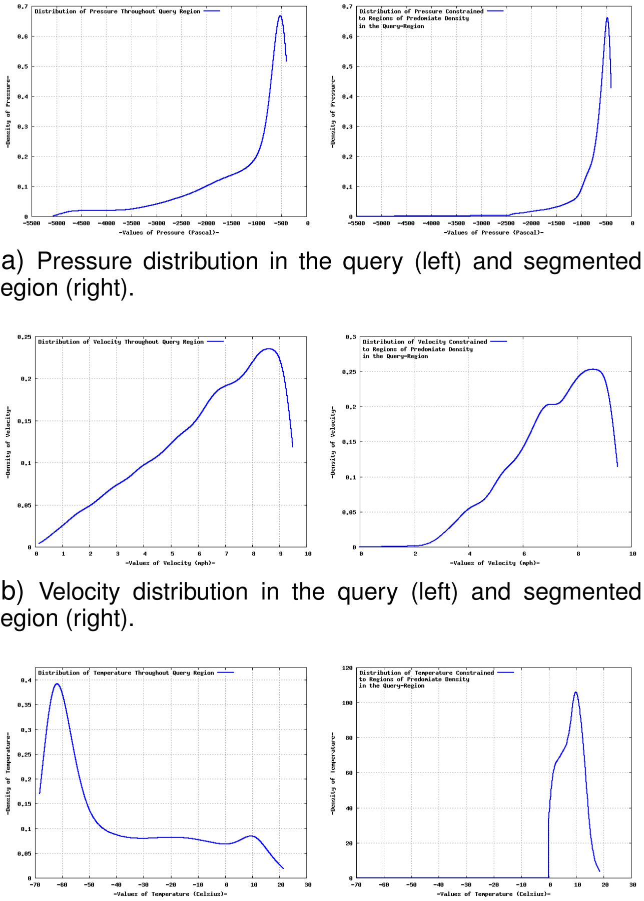
Fig. 4. These images depict the surface surrounding a query's solution set in (a), as well as the segmentation based on predominant distribution contributions within this query region. (b) shows regions where pressure (blue), velocity (green, and rendered transparent), and temperature (red) contribute most significantly to the joint distribution. Images (c) and (d) depict the result of refining the original query shown in (a) with the distribution information gathered for temperature obtained in (b).

solution set. In this image there are three well-defined visual regions of interest. The blue region corresponds to the areas where pressure's univariate distribution contributes the most to the query's joint distribution. Correspondingly, the green regions indicate areas where velocity plays the most significant influence in raising the values of the query's joint distribution. In comparison to these larger surfaces, we see a smaller red surface at the center of the query's solution set. This region corresponds to the areas where temperature plays the most significant role in contributing to query's joint distribution.

Analysis

We can interpret the significance of these visualizations by analyzing the univariate distribution of each variable as it is defined within the variable's segmented region, versus the query's solution set.

The left column in Fig. 5 shows the individual distributions of pressure (top), velocity (middle), and temperature (bottom) as they are found within the query's solution set. These histograms indicate that for this solution set Q, values above -1500 Pascal for pressure, above 5 mph for velocity, and below -50 degrees Celsius may play a predominant role in generating the query's joint distribution. We compare these distributions to those found in each variable's segmentation, shown at right in Fig. 5.



(a) Pressure distribution in the query (left) and segmented region (right).

(b) Velocity distribution in the query (left) and segmented region (right).

(c) Temperature distribution in the query (left) and segmented region (right).

Fig. 5. Based on the example in Section 4.1, these histograms illustrate (top to bottom) the univariate distributions of pressure, velocity, and temperature in the hurricane dataset as found through a query region (left column) and the regions where the respective variables are predominant in the approximated joint distribution (right column) for the query. Note that the two histograms (i.e. distributions) for temperature display vastly different modalities; in Section 4.1 we use this isolated range of data to direct refinement of query constraints.

In comparison to the distributions observed for pressure and velocity, the distribution for temperature's segmented region (bottom right in Fig. 5) demonstrates the isolation of a distinct modality from the distribution of temperature observed through the query's solution space (bottom left in Fig. 5). Specifically, the values for which temperature's univariate distribution most influences the joint distribution of the query are the range of values between 0 and 20 degrees Celsius. The strength of utilizing distribution-based segmentation is displayed in this example as the range of values from 0-20 degrees Celsius is obscured in temperature's observed univariate distribution in the query's solution space.

If the user was interested in further exploring this feature, the distribution of temperature's segmented region indicates a clear range of values for refining the query: $0 \leq \text{temperature} \leq 20$. Resubmitting the original query with this added constraint now isolates this region, as demonstrated by the surface shown

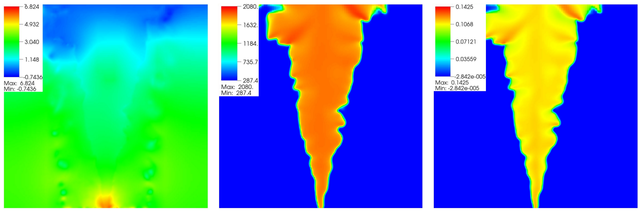


Fig. 6. Slices taken for three variables of the methane combustion dataset. Depicted are pressure (left), temperature (center), and Carbon Dioxide (right). We utilize these variables for analysis in Section 4.2.

in Fig. 4d. Alternatively, if the user was interested in excluding this feature from the query, the user could use the same range of values to exclude this feature as shown in Fig. 4c.

4.2 Methane Dataset

We apply our distribution and segmentation method to a query that analyzes a combustion dataset modeling a lean, premixed turbulent methane flame. This dataset incorporates 20 chemical species and 6 different physical properties (velocity, temperature, pressure, etc.). The simulation itself is simulated on a grid of size 300 x 300 x 300.

In this example, we utilize our method to analyze the data points selected from a query that constrains regions of high pressure, high temperature, and regions where the molecular concentration of CO_2 are above trace levels: specifically, pressure $\geq$ 2 atmospheres AND temperature $\geq$ 1000 Celsius AND $CO_2 \geq 1.0^{-8}$. The chemical species CO_2 is a final product of the combustion process of methane. The intent of this query is to extract data that can provide insight into how regions of increasing pressure and temperature within the combustion region propagate this chemical species throughout the flame.

The respective variables constrained by our query are depicted in Fig. 6. In this figure, we see slices through the combustion data's pressure (left), temperature (center), and CO_2 (right) concentration scalar fields.

Analysis

In the top image in Fig. 7, we see the minimum distribution surface for the query. In the next series of images we see the segmentation realized during the construction of the query's joint distribution. In the middle and bottom images in Fig. 7, the blue surface indicates the region where pressure contributes most to the query's joint distribution. Correspondingly, the red surface indicates the regions where CO_2 fundamentally increases the query's distribution. Wrapped between CO_2 and pressure, the segmented region for temperature is shown in green. For purposes of clarity in viewing the CO_2 region, the bottom image in Fig. 7 does *not* show the surface for pressure and temperature's surface (green) is rendered transparent.

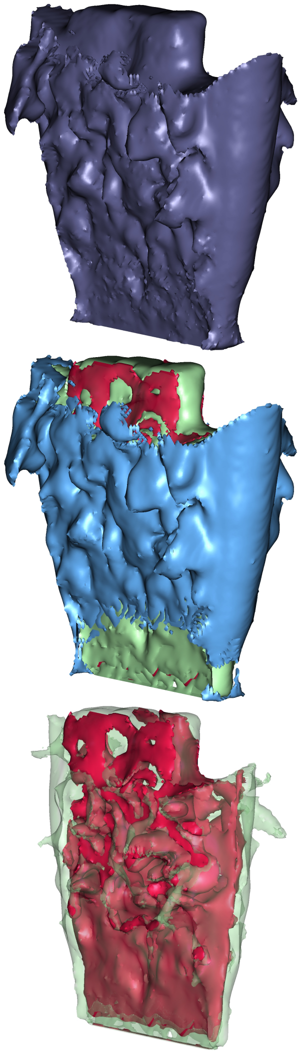
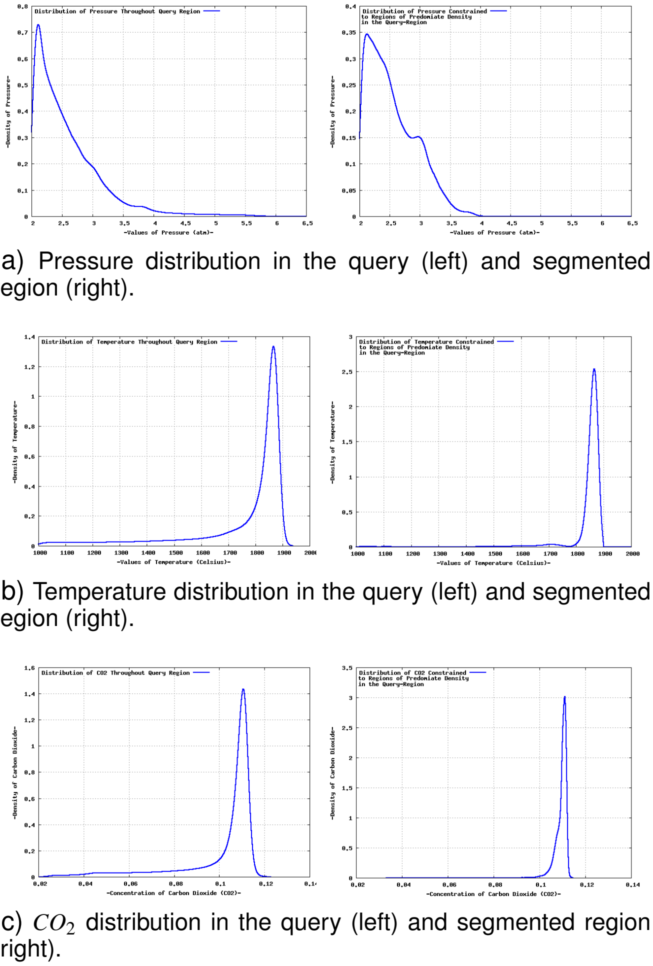


Fig. 7. These images visualize statistical data taken from a query that evaluates a methane combustion dataset. The image at top depicts a distribution-based surface that surrounds the query's solution set. The images at middle and bottom illustrate the segmentation for this query's joint distribution based on the maximal contribution of each constrained variable: pressure (blue), temperature (green), and CO_2 (red). The figure at bottom, which, for clarity, does *not* show pressure and uses a transparent surface for temperature, highlights the region where CO_2 (red) contributes most significantly to the query's solution.

We next observe the univariate distributions for each variable with respect to the query's solution space, and each variable's respective segmentation region. The left column in Fig. 8 shows individual distributions for pressure (top), temperature (middle), and CO_2 (bottom) as found within the query's solution set. We compare these distributions to those found in each variable's segmentation, at right in Fig. 8.

Though not as pronounced as the hurricane example, each variable's univariate distribution for their respective segmentation regions is subtly more refined. For example, a second modality (top right) has emerged for pressure in Fig. 8 at the range of 3 atmospheres. Also, temperature and CO_2 have no distribution for values less than 1800 Celsius and 0.1 respectively (unlike the distributions shown at left in Fig. 8).

In Fig. 9, we apply the strategy discussed in Section 3.2 and examine ranges of higher distribution values in the query's joint distribution. In this figure we see slices through the



(a) Pressure distribution in the query (left) and segmented region (right).

(b) Temperature distribution in the query (left) and segmented region (right).

(c) CO_2 distribution in the query (left) and segmented region (right).

Fig. 8. These histograms illustrate (top to bottom) the individual distributions of pressure, temperature, and concentration of Carbon Dioxide (CO_2) in the Methane Combustion dataset as found through the query region (left column) and the regions where the respective variables are predominant in the approximated joint distribution (right column). Note the histograms for each variable in the right column are more refined than those based on the entire query region (left column); specifically, major peaks for variables in the left column left have a higher relative distribution in the right column. Thus the blue (pressure), red (temperature), and green (Carbon Dioxide) colored regions in Fig. 7 (middle and bottom) are the areas where there is a narrow and refined range of values for each respective variable.

query solution's segmented regions oriented along the X (left) and Y (right) axis. Here gray regions indicate lower distribution regions in the query's joint distribution. Colored regions correspond to areas of higher joint distribution value where pressure (blue), temperature (green), and CO_2 (red) segmentation occurs. Note the reduced representation of pressure in these images. Contrariwise, note the predominance of temperature and CO_2 indicating these variables, and the values corresponding to each variable within its segmentation, play a more predominant role in the joint distribution.

5 Implementation and Performance

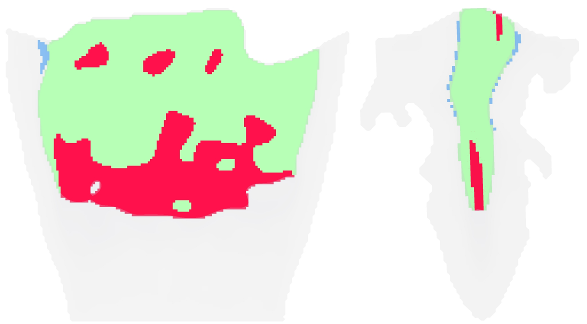


Fig. 9. Slices oriented along the X (left) and Y (right) axis of the segmented regions of the Methane dataset. Here segments for pressure (blue), temperature (green), and CO_2 (red) are depicted only for high distribution values. Grey regions indicate areas of lower distribution values in the query's solution set.

Query-Driven Visualization (QDV) demands components that are high-performing with respect to computation in order to support interactivity. Distribution estimations with KDE are of order $O(N^2)$ with a straightforward implementation and can be limiting upon overall performance for large N. There are methods for accelerating KDE calculations using up-front preprocessing, such as the Fast Gauss Transforms [44], Fast Multipole Method [20], and tree-based strategies [19]; however, the up-front processing is typically expensive and is amortized only if the KDE is evaluated frequently over fixed data values. For QDV applications, however, new KDE must be evaluated with every ad-hoc query, where the size of N for the KDE will typically be a small fraction of the total data. Paying a constant preprocessing cost for these acceleration methods (for a small and varying number of N) limits their utility for a QDV application. To accelerate our KDE implementation, we have thus turned to hardware acceleration.

We have implemented the GPU-based query engine presented by Gosink et al. [16], [17] to support rapid ad-hoc queries on large data sets. In this paper, we also compute the KDE distribution estimates and surfaces for visualization on the GPU. Our KDE computation takes as input a list of data samples that pass the query. We launch a GPU thread per data sample to evaluate the multivariate Gaussian kernel in (4). By keeping the query solution on the GPU we can exploit the inherent parallelism of the graphics hardware to accelerate the KDE computation, in lieu of transferring the data back to main memory for CPU computation.

In measuring the performance of our implementation, there are two factors that affect our timings: increasing size for query solutions sets (i.e. decreasing the query's selectivity[1]), and increasing the number of variables for the joint distribution computation. We analyze these two metrics independently by analyzing increasingly larger subsets of data, in conjunction with queries that constrain an increasing number of variables. The performance for this test are presented in Table 1. The performance times are based on the hurricane dataset which consists of 8.1 million cells mapped to a 300 X 300 x 90 uniform grid.

[1] Query selectivity is the number of dataset records selected by the query versus the total number of available records in the data.

TABLE 1
This table depicts the performance times, in seconds, for our gpu-based distribution estimation implementation. The axis are decreasing selectivity vs increasing variable count. Times include the time to access all raw data from CPU-memory, load the data to the GPU, compute the distribution, determine the segmentation (for multivariate queries), and write the solution back to CPU memory.

Variables Queried	select 1% (seconds)	select 2.5% (seconds)	select 5% (seconds)	select 10% (seconds)
1	1.09	8.3	27.6	109.0
2	2.2	16.4	57.2	227.0
3	3.1	23.02	78.7	301.0
4	3.8	30.4	97.53	386.0

From Table 1 we observe that our implementation follows an expected performance trend for an $O(N^2)$ algorithm. Additionally, we see that increasing the number of variables utilized to construct a distribution scales with an expected linear growth curve. In practice, we have found that these processing times are not prohibitive – once the KDEs have been computed, users are able to interactively explore different distribution surfaces and the multivariate segmentation for analysis purposes.

6 Conclusion

Herein, we have presented a method that uses a statistical framework to estimate the underlying distribution of data within a query's solution. This approach allows for the construction of boundary surfaces for query regions based upon the behaviors of one or more variables. Furthermore, users are able to directly visualize the structure of the query in terms of the multivariate distribution, or through a segmentation formed by the univariate distribution estimations. The utility of these methods in a QDV setting has been demonstrated across two scientific datasets.

Our preliminary research indicates that additional statistical measures might be useful in characterizing and visualizing multivariate behavior within query regions. For example, we have demonstrated a segmentation based on maximal-contributions from a collection of variables. However, segmentation based on mean and furthest outlying variable (i.e. the variable furthest from the mean of variable distribution values) have also displayed promising results. Furthermore, we plan to investigate the use of textures on each variable's segmented surface to convey additional statistical information.

We are also actively researching how to effectively relate analysis results obtained from a query-selected region of the dataset to the whole dataset: e.g. *how* statistically significant is the selected region in comparison to other regions of the dataset, or even the whole dataset? To this degree we are investigating strategies that can employ information uncertainty into QDV so that the neighborhood surrounding a query's selection can be taken into consideration by our statistical framework.

Finally, we are working on implementing this statistical framework into the VisIt software package [8]. In our implementation, users will load multiple variables from a single database into a VisIt viewer. Users will then express and refine queries by adjusting individual variable constraints using interactive sliders. Assisted by a GPU-accelerated engine, our statistical framework will facilitate the visualization and exploration of the various distribution surfaces and segmented regions within the query's solutions space.

Acknowledgments

This work was supported by Lawrence Berkeley National Laboratory and by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231 through the Scientific Discovery through Advanced Computing (SciDAC) program's Visualization and Analytics Center for Enabling Technologies (VACET). We thank our colleagues in the Institute for Data Analysis and Visualization (IDAV) at UC Davis for their support during the course of this work. Additionally, the authors thank Bill Kuo, Wei Wang, Cindy Bruyere, Tim Scheitlin, and Don Middleton of the U.S. National Center for Atmospheric Research (NCAR), and the U.S. National Science Foundation (NSF) for providing the Weather Research and Forecasting (WRF) Model simulation data of Hurricane Isabel. Data for the Methane simulation was provided by John Bell and Marc Day, Center for Computational Science and Engineering, Lawrence Berkeley National Laboratory. We also thank our reviewers for helping to improve this paper. Last a warm acknowledgement goes to George Roussas who inspired this work in earlier conversations.

References

[1] John C. Anderson, Luke J. Gosink, Mark A. Duchaineau, and Kenneth I. Joy. Feature identification and extraction in function fields. In *EuroVis 2007*, pages 195–201, May 2007.

[2] Chandrajit L. Bajaj, Valerio Pascucci, and Daniel R. Schikore. The contour spectrum. In *Proc. of IEEE Visualization*, pages 167–173, 1997.

[3] David C. Banks and Stephen Linton. Counting cases in Marching Cubes: Toward a generic algorithm for producing substitopes. In *Proc. of IEEE Visualization*, pages 51–58, October 2003.

[4] Richard A. Becker and William S. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987.

[5] E. Wes Bethel, Scott Campbell, Eli Dart, Kurt Stockinger, and Kesheng Wu. Accelerating network traffic analysis using query-driven visualization. In *Proc. of IEEE Symposium on Visual Analytics Science and Technology*, pages 115–122, October 2006.

[6] David M. Butler, James C. Almond, R. Daniel Bergeron, Ken W. Brodlie, and Robert B. Haber. Visualization reference models. In *Proc. of IEEE Visualization*, pages 337–342, 1993.

[7] Hamish Carr, Duffy Brian, and Denby Brian. On histograms and isosurface statistics. *IEEE Trans. on Visualization and Computer Graphics*, 12(5):1259–1266, 2006.

[8] Hank Childs, Eric S. Brugger, Kathleen S. Bonnell, Jeremy S Meredith, Mark Miller, Brad J Whitlock, and Nelson Max. A contract-based system for large data visualization. In *Proceedings of IEEE Visualization 2005*, pages 190–198, 2005.

[9] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *Proc. of IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24:603–619, 2002.

[10] Roger A. Crawfis and Nelson Max. Texture splats for 3d scalar and vector field visualization. In *Proc. of IEEE Visualization*, pages 261–266, 1993.

[11] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. on Information Theory*, 21(1):32–40, 1975.

[12] Bogdan Georgescu, Ilan Shimshoni, and Peter Meer. Mean shift based clustering in high dimensions: A texture classification example. *Computer Vision, IEEE International Conference on*, 1:456, 2003.

[13] Markus Glatter, Jian Huang, Sean Ahern, Jamison Daniel, and Aidong Lu. Visualizing temporal patterns in large multivariate data using modified globbing. *Proc. of IEEE Trans. on Visualization and Computer Graphics*, 14(6):1467–1474, 2008.

[14] Markus Glatter, Jian Huang, Jinzhu Gao, and Colin Mollenhour. Scalable data servers for large multivariate volume visualization. *IEEE Trans. on Visualization and Computer Graphics*, 12(5):1291–1298, 2006.

[15] Luke J. Gosink, John C. Anderson, E. Wes Bethel, and Kenneth I. Joy. Variable interactions in query-driven visualization. *Proc. of IEEE Trans. on Visualization and Computer Graphics*, 13(6):1400–1407, 2007.

[16] Luke J. Gosink, John C. Anderson, E. Wes Bethel, and Kenneth I. Joy. Query-driven visualization of time-varying adaptive mesh refinement data. *Proc. of IEEE Trans. on Visualization and Computer Graphics*, 14(6):1715–1722, 2008.

[17] Luke J. Gosink, Kesheng Wu, E. Wes Bethel, John D. Owens, and Kenneth I. Joy. Bin-Hash Indexing: A parallel method for fast query processing. Technical Report LBNL-729E, Lawrence Berkeley National Laboratory, 2008. http://www.vis.lbl.gov/Publications/2008/LBNL-729E.pdf.

[18] Luke J. Gosink, Kesheng Wu, E. Wes Bethel, John D. Owens, and Kenneth I. Joy. Data parallel bin-based indexing for answering queries on multi-core architectures. In *Conference on Scientific and Statistical Database Management*, volume 5566, pages 110–129, June 2009.

[19] Alexander G. Gray and Andrew W. Moore. Rapid evaluation of multiple density models. In *Proc. of the 9th International Workshop on Artificial Intelligence and Statistics*, 2003.

[20] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73(2):325–348, December 1987.

[21] Hans-Christian Hege, Martin Seebass, Detlev Stalling, and Malte Zöckler. A generalized Marching Cubes algorithm based on non-binary classifications. Technical Report SC-97-05, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 1997.

[22] Yannis Ioannidis. The history of histograms (abridged). In *Proc. VLDB*, pages 19–30, 2003.

[23] Heike Jänicke, Michael Böttinger, and Gerik Scheuermann. Brushing of attribute clouds for the visualization of multivariate data. *IEEE Trans. on Visualization and Computer Graphics*, 14(6):1459–1466, 2008.

[24] Tao Ju, Frank Losasso, Scott Schaefer, and Joe Warren. Dual contouring of Hermite data. *ACM Trans. on Graphics*, 21(3):339–346, 2002.

[25] Joe Kniss, Gordon Kindlmann, and Charles Hansen. Multidimensional transfer functions for interactive volume rendering. *Proc. of IEEE Trans. on Visualization and Computer Graphics*, 8(3):270–285, 2002.

[26] Christian Ledergerber, Gaël Guennebaud, Miriah Meyer, Moritz Bächer, and Hanspeter Pfister. Volume MLS ray casting. *IEEE Trans. on Visualization and Computer Graphics*, 14(6):1372–1379, 2008.

[27] Lars Linsen, Tran Van Long, Paul Rosenthal, and Stephan Rosswog. Surface extraction from multi-field particle volume data using multi-dimensional cluster visualization. *In Proc. of IEEE Trans. on Visualization and Computer Graphics*, 14(6):1483–1490, 2008.

[28] Z. Liu, W. Chen, K.Q. Huang, and T.N. Tan. A probabilistic framework based on KDE-GMM hybrid model (KGHM) for moving object segmentation in dynamic scenes. In *Workshop on Visual Surveillance*, 2008.

[29] W. E. Lorensen and H. E. Cline. Marching Cubes: A high resolution 3D surface construction algorithm. In *Proc. of SIGGRAPH*, pages 163–169, 1987.

[30] Claes Lundstrom, Patric Ljung, and Anders Ynnerman. Local histograms for design of transfer functions in direct volume rendering. *IEEE Trans. on Visualization and Computer Graphics*, 12(6):1570–1579, 2006.

[31] Klaus Mueller, Torsten Müller, J. Edward Swan II, Roger Crawfis, Naeem Shareef, and Roni Yagel. Splatting errors and antialiasing. *IEEE Trans. on Visualization and Computer Graphics*, 4(2):178–191, 1998.

[32] Gregory M. Nielson and Richard Franke. Computing the separating surface for segmented data. In *Proc. of IEEE Visualization*, pages 229–233, October 1997.

[33] Emanuel Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

[34] Murray Rosenblatt. Remarks on some nonparametric estimates of a density function. In *Annals of Mathematical Statistics*, volume 27, page 832835, 1956.

[35] Carlos E. Scheidegger, John M. Schreiner, Brian Duffy, Hamish Carr, and Cláudio T. Silva. Revisiting histograms and isosurface statistics. *IEEE Trans. on Visualization and Computer Graphics*, 14(6):1659–1666, 2008.

[36] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall/CRC, April 1986.

[37] Kurt Stockinger, John Shalf, Kesheng Wu, and E. Wes Bethel. Query-driven visualization of large data sets. In *Proc. of IEEE Visualization*, pages 167–174, October 2005.

[38] Lee Westover. Interactive volume rendering. In *Proc. of the Workshop on Volume Visualization*, pages 9–16, 1989.

[39] Lee Westover. Footprint evaluation for volume rendering. In *Proc. of SIGGRAPH*, pages 367–376, 1990.

[40] Pak Chung Wong and R. Daniel Bergeron. Multiresolution multidimensional wavelet brushing. In *Proc. of IEEE Visualization*, pages 141–148, 1996.

[41] K. Wu, S. Ahern, E. W. Bethel, J. Chen, H. Childs, E. Cormier-Michel, C. G. R. Geddes, J. Gu, H. Hagen, B. Hamann, W. Koegler, J. Laurent, J. Meredith, P. Messmer, E. Otoo, V. Perevoztchikov, A. Poskanzer, Prabhat, O. Rübel, A. Shoshani, A. Sim, K. Stockinger, G. Weber, and W.-M. Zhang. FastBit: Interactively Searching Massive Data. *Journal of Physics Conference Series*, 2009.

[42] Kesheng Wu, Wendy S. Koegler, Jacqueline Chen, and Arie Shoshani. Using bitmap index for interactive exploration of large datasets. In *Proc. of Scientific and Statistical Database Management*, pages 65–74, 2003.

[43] Kesheng Wu, Ekow J. Otoo, and Arie Shoshani. On the performance of bitmap indices for high cardinality attributes. In *Proc. of VLDB*, pages 24–35, 2004.

[44] Changjiang Yang, Ramani Duraiswami, Nail A. Gumerov, and Larry Davis. Improved fast Gauss transform and efficient kernel density estimation. In *Proc. of IEEE International Conference on Computer Vision*, volume 1, page 464, 2003.

[45] Xiaotong Yuan, Bao-Gang Hu, and Ran He. Agglomerative mean-shift clustering via query set compression. In *In Proc. of the SIAM International Conference on Data Mining*, pages 221–232, 2009.

[46] Xiang Zhang and Jie Yang. Moving object detection based on shape prediction. *Journal of the Optical Society of America*, 26(2):342–349, 2009.