Computers & Graphics 41 (2014) 62–71

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Special Section on Uncertainty and Parameter Space Analysis in Visualization

Visualizing likelihood density functions via optimal region projection

Hal Canary [a], Russell M. Taylor II [d,*], Cory Quammen [a], Scott Pratt [b], Facundo A. Gómez [b], Brian O'Shea [b], Christopher G. Healey [c]

[a] Department of Computer Science, University of North Carolina at Chapel Hill, United States
[b] Department of Physics and Astronomy, Michigan State University, United States
[c] Department of Computer Science, North Carolina State University, United States
[d] Department of Computer Science, University of North Carolina at Chapel Hill, CB #3175, Sitterson Hall, Chapel Hill, NC 27599-3175, United States

ARTICLE INFO

Article history:
Received 15 July 2013
Received in revised form
9 February 2014
Accepted 26 February 2014
Available online 12 March 2014

Keywords:
Uncertainty
Parameter space analysis
Visualization
Likelihood density function

ABSTRACT

Effective visualization of high-likelihood regions of parameter space is severely hampered by the large number of parameter dimensions that many models have. We present a novel technique, Optimal Percentile Region Projection, to visualize a high-dimensional likelihood density function that enables the viewer to understand the shape of the high-likelihood region. Optimal Percentile Region Projection has three novel components: first, we select the region of high likelihood in the high-dimensional space before projecting its shadow into a lower-dimensional projected space. Second, we analyze features on the surface of the region in the projected space to select the projection direction that shows the most interesting parameter dependencies. Finally, we use a three-dimensional projection space to show features that are not salient in only two dimensions. The viewer can also choose sets of axes to project along to explore subsets of the parameter space, using either the original parameter axes or principal-component axes. The technique was evaluated by our domain-science collaborators, who found it to be superior to their existing workflow both when there were interesting dependencies between parameters and when there were not.

1. Introduction

A basic question in any field of science is how to choose the theory that best fits the evidence. Given a set of experimental observations, how does one find the model that best fits the data? And after choosing a model, how does one quantify the level of confidence in that model?

This research addresses the specific case of comparing the explanatory power of variations on a single model where those variations can be described by a list of model parameters that can vary continuously. In statistics, the term *likelihood* ($\mathcal{L}$) is used to refer to the probability of a set of parameter values given a set of observations. The likelihood of a set of parameter values is calculated by comparing a set of *model outputs* with observed quantities.

We refer to the set of possible parameter values as *parameter space*. Because the integral of finite values within a zero-radius sphere is zero, the probability that any particular point in parameter space is correct—even the location with maximum likelihood—is zero. Thus, our collaborators are interested in identifying the shape of the high-likelihood region of parameter space, which tells them how the parameters interact in the region of highest likelihood.

We collaborate with researchers studying galaxy formation and relativistic particle collisions who run large ensemble simulations to try and determine the most-likely parameter values for models of the fundamental behavior of the universe.

The models under study by our collaborators have between 5 and 20 parameters. For the larger models, uniform sampling of the entire parameter space with a grid fine enough to reveal important details is not feasible given the memory sizes and computational power available to them. Even if it were, direct visualization of *n*-D results cannot be done on a 2D or 3D display without some sort of projection. This research presents novel visualization tools that display the shape and extent of high-likelihood regions of parameter space. These tools make features salient that could not be seen using previous techniques.

2. Background and related work

We first describe the background mathematics used by our technique to project likelihood from the high-dimensional parameter space into a lower-dimensional visualization space, and then describe existing techniques for doing this projection and display.

[*] Corresponding author. Tel.: +1 919 590 6001; fax: +1 919 590 6105.
E-mail address: taylorr@cs.unc.edu (R.M. Taylor II).
URL: http://www.cs.unc.edu/~taylorr (R.M. Taylor II).

http://dx.doi.org/10.1016/j.cag.2014.02.005
0097-8493/© 2014 Elsevier Ltd. All rights reserved.

2.1. Likelihood density function

Many problems in both the natural and social sciences now involve large scale models characterized by numerous parameters. These models are then often compared to experimental data sets, which in some cases are distilled from peta-scale observations. This results in a high-dimensional scalar field (one dimension per parameter) that describes how likely it is that the parameters associated with each point match experimental results. It is this scalar field whose properties we display.

Perhaps the most common method for determining the optimal values of the set $\overrightarrow{x}$ of p model parameters $x_1 \ldots x_p$ is using comparison to a set $\overrightarrow{y}$ of measurements $y_1 \ldots y_M$ to calculate the implausibility, χ^2, as a function of $\overrightarrow{x}$ where $\chi^2(\overrightarrow{x}) \geq 0$ describes the “poorness” of the fit for a specific point in parameter space and is zero for a perfect fit,

$$\chi^2(\overrightarrow{x}) \equiv \sum_a \frac{(y_a^{(\text{mod})}(\overrightarrow{x}) - y_a^{(\text{exp})})^2}{\sigma_a^2}, \tag{1}$$

where $a \in 1 \ldots \text{M}$ sums over all measurements. $y_a^{(\text{mod})}$ refers to the ath measurement computed by the model, and $y_a^{(\text{exp})}$ to the corresponding experimental measurement. σ_a is a measure of the uncertainty for comparing the model to the experiment, and can come from uncertainties in the model or from expected experiment measurement errors. As σ_a approaches zero, the implausibility approaches infinity for model measurements that differ at all from the experiment.

If the uncertainty involved in comparing the model to the data is normally distributed, and if there is no prior information about the parameters (a flat prior distribution), Bayes theorem tells us that the likelihood that the point in parameter space could reproduce the data is given by the likelihood density function $\mathcal{L}$:

$$\mathcal{L}(\overrightarrow{y} \,|\, \overrightarrow{x}) \sim \exp\left\{ -\frac{\chi^2(\overrightarrow{x})}{2} \right\}. \tag{2}$$

In some cases, one is interested in only the point of minimum χ^2 (maximum likelihood), but a much more appropriate goal is to understand the entire distribution $\mathcal{L}(\overrightarrow{y} \,|\, \overrightarrow{x})$, so that one knows not only the most likely point, but understands the range and distribution of likely values of $\overrightarrow{x}$.

2.2. Markov chain Monte-Carlo sampling

The standard method of computing marginal probability from a likelihood density function is to use a Markov chain Monte-Carlo (MCMC) sampling of that function. MCMC algorithms have the property that they produce point samples whose equilibrium spatial density is proportional to the local likelihood density. This has proven to be an effective way to approximate integrals over the multidimensional domain [1].

This produces a large set of points in parameter space, each of which has an associated likelihood value, whose spatial density is proportional to local likelihood. The point density can be used to integrate the likelihood by counting the number of points that fall within each bin in a spatial lattice; these counts are proportional to the integrated likelihood within each bin. These points can be projected into a lower-dimensional visualization space before being binned.

Formally, let $\mathbb{R}^p$ be the p-dimensional space of real numbers. Let $\mathcal{L} : \mathbb{R}^p \to \mathbb{R}$ be the likelihood density function over a domain with p continuous parameters (the likelihood density at the specified coordinate). If $T : \mathbb{R}^p \to \mathbb{R}^r$ is a projection function from the p-dimensional parameter space to an r-dimensional subspace with $p > r$, the MCMC samples can be analyzed to estimate the integral of likelihood within the subspace. This calculation is performed by applying T to each of the MCMC samples, and then approximating the density in $\mathbb{R}^r$ by binning the results.

For the likelihood density functions that we received from our collaborators, a few million samples were enough to approach the equilibrium distribution with a resolution finer than the scale of the features of interest.

2.3. Related work

There are a number of existing dimension-reduction techniques including projection-based methods, dimension selection, stress-based optimization methods, multidimensional scaling and others [2]. Experimental comparison among several techniques is provided in [3]. We present these techniques below and show how our approach builds on and extends them to address our collaborators' needs.

2.4. Orthographic all-point projections

The left side of Fig. 1 shows two standard projections: histograms and the scatter-plot matrix.

One example of such a projection function is $T(\overrightarrow{x}) = (x_i)$. The output values of this projection are plotted in a histogram that shows the relative likelihood of each value of the ith parameter.

Another example of a projection function is $T(\overrightarrow{x}) = (x_i, x_j)$, which will produce a *scatter plot matrix* when all combinations of $i \neq j$ are considered. When the number of points is large, the scatter plots can be replaced with density plots to avoid overplotting, as is done in Fig. 1.

A limitation of such projections that send all points into the subspace, integrating the likelihood during projection, is that they provide only independent statistical information about the projected parameters and lose information about how the parameters are related to each other in the original space. To demonstrate this, consider the H function shown in Fig. 2,

$$H(x_1, x_2) = \begin{cases} \exp\{-x_1^2\}\exp\{-(2\pi x_2 \exp\{-x_1^2\})^2\} \\ \quad \text{if } |x_1| \leq \sqrt{\pi}. \\ 0 \\ \quad \text{if } |x_1| > \sqrt{\pi}, \end{cases}$$

a two-dimensional function with a clear maximum at (0, 0) in the original 2D space. When projected onto the x_1-axis, each value of the x_1 parameter within the range $[-\sqrt{\pi}, \sqrt{\pi}]$ has equal probability:

$$\begin{aligned} H_{\text{proj.}}(x_1) &= \int_{-\infty}^{\infty} H(x_1, x_2)\, \mathrm{d}x_2 \\ &= \begin{cases} \dfrac{1}{2\sqrt{\pi}} & \text{if } |x_1| \leq \sqrt{\pi} \\ 0 & \text{if } |x_1| > \sqrt{\pi} \end{cases} \end{aligned}$$

If one is only interested in the possible values of x_1, independently of x_2, then this projection will effectively answer that question. However, our collaborators are also interested in understanding which combinations of model parameters are most likely. That is, we are tasked with communicating facts about the original function in its original domain, using a representation in a projected space.

This problem can also occur in 2D projections of higher-dimensional objects. Some relationships between parameters are much easier to discern in 3D than in 2D projections. For example, the cup function in Fig. 1,

$$C(x_1, x_2, x_3) = \exp\{-x_1^2 - x_2^2 - x_3^2 - 100(x_1^2 + x_2^2 - x_3)^2\},$$

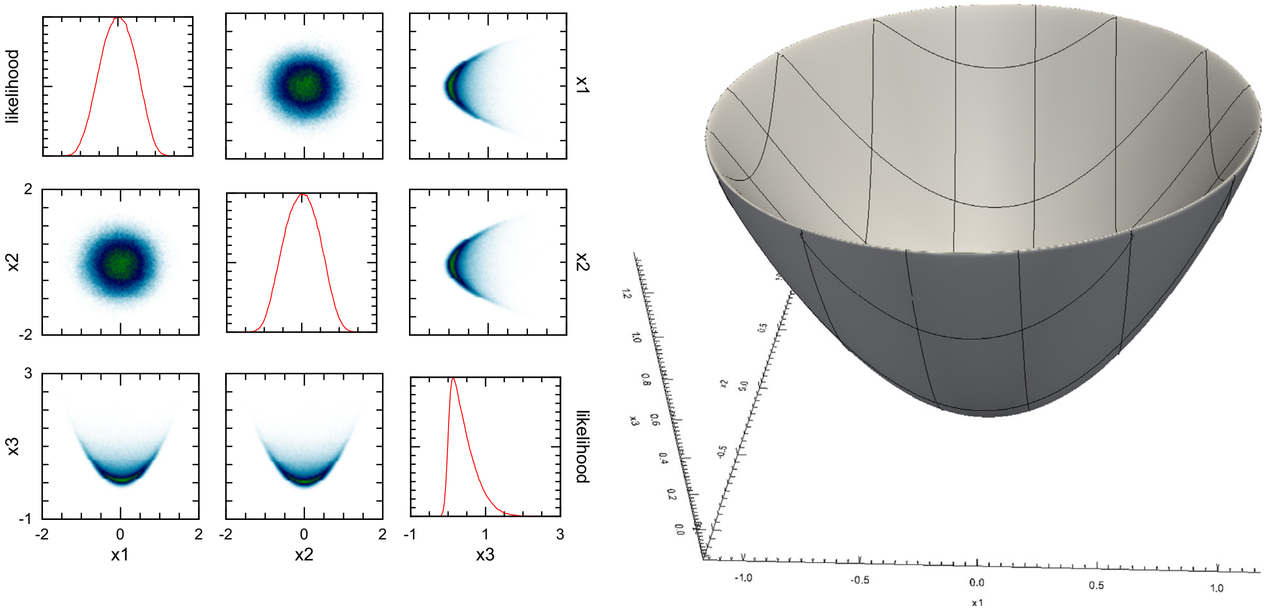

Fig. 1. Visualizing the cup function as a scatterplot matrix (left) and as an iso-surface of the projected region in 3D (right).

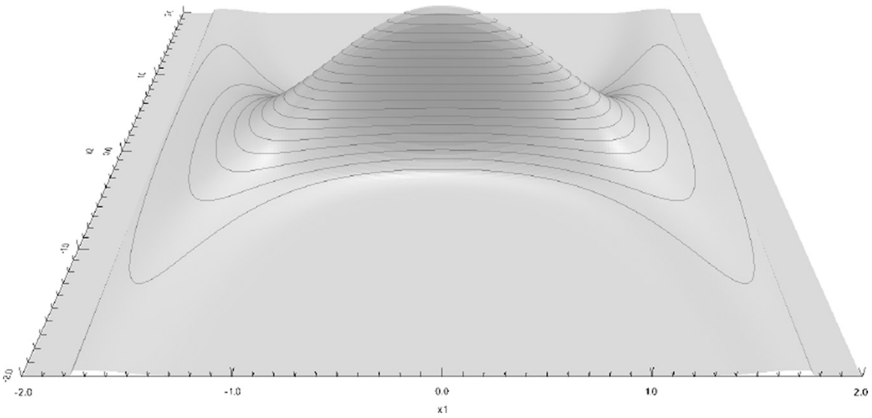

Fig. 2. $z = H(x, y)$.

has a maximum region that looks like a cup or bowl. If we project and plot the samples in a scatterplot matrix, we do not see the cup shape, only a crescent. Seeing this shape requires a 3D (rather than 2D) projection.

Of course, this problem continues all the way up to the dimension of the original space. However, with each increase in display dimension, the viewer can immediately see more complex dependencies among parameters.

These examples hint at the information lost when projecting all points from high-dimensional spaces down to a 2D space and displaying the scatter-plot matrix of all pairs. After using such plots, our collaborators asked explicitly for help with the issue of locating “banana shapes” in high dimension caused by unexpected dependencies among parameters. After several rounds of discussions and exploration of potential mathematical descriptions for a “banana shape”, we developed the approach described here.

2.5. Other projection approaches

One possible approach is to perform n-dimensional clustering to identify regions of high probability points in n-D space, assuming the standard issues with this type of clustering could be overcome (e.g. all points are far form one another along the majority of their dimensions, producing poor distance discrimination). For example, recent advances in multidimensional scaling (MDS) enable the projection and interactive display of data sets with millions of points, such as the ones generated by our collaborators [4]. This approach enables the visual detection of clusters in the data, keeping nearby points in the high-dimensional space nearby in the projection. The resulting projections do not attempt to maintain relationships among individual dimensions, however, so they can produce severe shape distortions in the projected region.

Perhaps more importantly, in this project our collaborators are not asking specifically about cluster locations and relationships between them. Instead, they need to understand the shape of the high probability regions in n-D space, and locating interesting shape features. We initially developed MDS-based techniques to allow our collaborators to study the early phase of the MCMC process as it searches for equilibrium, but they were not well suited to the analysis phase where our collaborators are looking for relationships among multiple dimensions.

Another possibility is the Dimstiller system [5], which provides flexible, interactive, multi-window displays for exploration of high-dimensional parameter spaces. They found that 2D projections along principal-component directions produced display of structure not seen in projections done using multidimensional scaling and original-axis directions. Our approach automatically

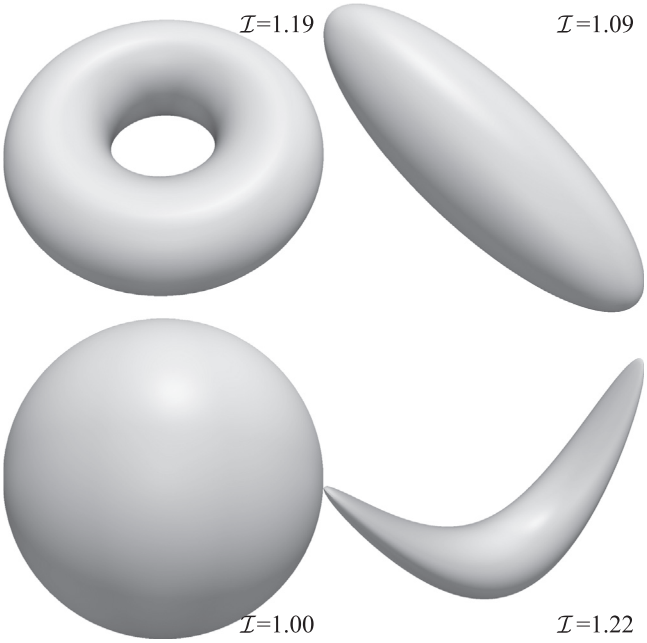

Fig. 3. Comparison of the normalized shape index $\mathcal{I}$ for four common shapes.

searches a broader set of projection directions to guide the user to regions of interest, in addition to providing 3D display of the selected axes. These techniques could be readily added to Dimstiller and other such systems as an additional workflow and display type. It is not clear how one would implement high-dimensional region selection in Dimstiller, but it could be a preprocess.

Projection Pursuit [6] selects an axis of projection given an "interestingness" function; our approach extends this to a 3D multi-index pursuit with an "interestingness" function appropriate to detecting parameter dependencies.

2.6. *n-Dimensional visualization techniques*

We and others have used Parallel Coordinates (PC) techniques to display relationships among large sets of parameters [7,8]. PC places pairs of axes representing individual data attributes in spatial proximity to one another [9]. A data sample is positioned at its attribute value location on each axis. Connecting these locations produces a line that visualizes the sample. Viewers can identify common polylines, which correspond to clusters of data samples with similar attribute values. One possibility would be to select high probability points, then plot them in PC where each axis corresponds to one hyper-dimension. This could enable a viewer to identify points with common polylines, indicating a set of high probability points in a common hyper-region. The detection of relationships in such displays relies heavily on the axis ordering, however, and tracing curves across several intervening axes make it difficult to recognize these relationships. Because PC performs clustering, it also suffers from the fact that clusters do not necessarily answer our collaborators' questions about shape understanding and feature detection. Indeed, we initially tried a PC-based approach. Issues our collaborators encountered within that approach motivated us to pursue the projection technique discussed in this paper.

Topology-based approaches such as Landscape Profiles [10] and Topological Spines [11] are very effective for the display of the relative sizes and symmetries between different high-likelihood regions in the high-dimensional parameter space. When we implemented these approaches and ran them on our collaborators data sets, we found that there was a single region of high likelihood. Again, these methods do not attempt to maintain between-axis consistency in the projection, so they were not directly usable to explore the questions our collaborators had with respect to the shape of the high-likelihood region and how that informed parameter dependencies.

The XGobi system [12] provides a flexible, interactive environment to explore high-dimensional data sets. It includes the ability to explore principal-component and custom mixtures of the original parameter dimensions, and it allows selection of pairs and triples of dimensions for visualization. Our work fits into their "finding Gestalt" task; it provides an automatic way for our collaborators to estimate useful projection directions and augments it with a pre-filtering of the data in high-dimensional space that removes irrelevant points prior to projection. These techniques could be easily added to XGobi or similar systems, and they would benefit from such systems' ability to animate the transition from one projection to another.

Glyphs are often used for multivariate visualization [13]. In a dataset with high spatial dimensionality, however, even multivariate glyph approaches need to project data elements into the display space (e.g. onto a 2D place or into a 3D volume). Glyphs are normally used to visualize multiple attribute values, and usually after a spatial embedding has been defined. For example, a common approach would use properties of color, texture, and motion to visualize multiple attribute values attached to each data element [14,15]. Although our data elements have only a single likelihood attribute, if more attributes were provided, a multivariate glyph approach could be considered to represent these multiple values. This would still require a way to project the *n*-D elements into 3D, however.

3. Methodology

Our method does three things beyond the standard all-points 2D projections that are common in our collaborators' workflow. We summarize the approach here and provide details in the following sections.

First, we select a high-likelihood region in the original high-dimensional space prior to projection into the lower-dimensional space. This avoids the information loss incurred when projecting first and then selecting a high-likelihood region, enabling our method to display important parameter relationships in the original space.

Second, we project into 3D rather than 2D to preserve as much information about dependencies as can be effectively comprehended by the human visual system. Because the human visual system is attuned to perceiving surfaces rather than volumetric data, we designed a visualization that produces a surface in three dimensions. (Volume display provides the ability to see inside the volume but hampers shape perception due to a lack of occlusion, causing inability to clearly perceive relative depths.)

Third, we select an initial projection by maximizing a metric that prefers axis sets that have more interesting dependencies between the parameters. Rather than seeking any particular shape ("banana"), this metric penalizes simpler shapes that can be explained purely in terms of covariance and selects the ones whose relationships are not simple to describe ("not an orange"). This results in the detection of the most-interesting shape, whether it is an apple or a pear or a strawberry. The scientist is free to explore all sets of three parameter axes, and we provide an interface for them to select among them. Additionally, the axis-selection procedure can explore the space of orthogonal linear combinations of axes, locating interesting parameter dependencies that are not present along any axis-aligned or principal-component projection.

3.1. Percentile region selection

Although it can be used in other ways, we will describe our algorithm within the context of our collaborators' workflow to provide a concrete example. We are provided by each collaborator with a function $F(\overrightarrow{x})$ which, given a point $\overrightarrow{x}$ in parameter space, computes the likelihood density value, $\mathcal{L}$, of the model at that point. When the simulation is fast enough, it is used directly. For slower simulations, a Gaussian mixture model is used as a rapidly computable emulator.

We first apply a Metropolis–Hastings [16,17] approach to perform MCMC-based integration in the parameter space. We run MCMC integration for one million steps, each step calling F, to produce one million points in parameter space (each of which is annotated by its likelihood density). This step takes about 20 min on a single processor for the emulator used in the galaxy-formation study, but can be linearly parallelized; it is a pre-process that is run once for each model.

We then sub-sample the points down to ten thousand for the purpose of analysis and visualization. (We cannot simply use the first 10,000 MCMC steps because the process will not have converged.) Because of the properties of MCMC integration, the local density of points is proportional to the local likelihood density $\mathcal{L}$. Each point remains annotated with its actual value of $\mathcal{L}$.

We first select the region of parameter space containing the 95% of sample points with highest values ($R \subset \mathbb{R}^p$). (The particular percentile found is a user-selectable level that defaults to 95%.) The pointwise labeling with $\mathcal{L}$ lets us select points within the 95th percentile without having to estimate local point densities in high dimensional parameter space.

To select the points, we first find the 5% order statistic on likelihood, which is the likelihood threshold above which 95% of the points in our sample lie. This threshold value is the greatest lower bound of likelihood density function values within R. After discarding the points below the threshold, we project the remaining samples into three dimensions using a linear orthonormal projection. Around these points we compute and display a tight-fitting surface as described in the next section.

3.2. Finding the boundary surface

The Boundary portion of the PercentileRegionProjection algorithm can use any method that returns a tight-fitting surface around X_{proj}, a set of points in space. We initially convolved the points with a Gaussian to produce a smooth volumetric density distribution and then computed an iso-surface of the resulting density field. This method was very computationally expensive for the large number of sample points and a Gaussian that is the size of the expected features in the data. It also depended on the specification of a threshold value for the iso-surface that changes the tightness of fit. Finally, the resulting surface did not pass through the boundary of the points.

An alternative method that was used to produce the figures in this paper calculates the three-dimensional Delaunay triangulation of X_{proj}. This results in a set of tetrahedra, the outside surface of which is the convex hull of X_{proj}. Because we want a tight-fitting surface rather than the convex hull, we remove tetrahedra whose largest edge is longer than the scale set by the user as the smallest feature of interest in the resulting surface. After removing these tetrahedra, the surface is the boundary of the resulting simplicial complex.

```
PercentileRegionProjection ($\mathcal{L}, N, c, T$).
    // $\mathcal{L}$ = the likelihood function to be visualized.
    // $N$ = the number of samples needed to sample $\mathcal{L}$.
    // $c \in [0,1]$ = the percentile to be visualized.
    // $T \in \mathbb{R}^{3\times p}$ = projection matrix.
    $X = \text{Metropolis-Hastings}(\mathcal{L}, N)$ // $X \subset \mathbb{R}^p, |X| = N$
    $v = \text{FindOrderStatistic}(\{\mathcal{L}(x) : x \in X\}, (1-c)N)$
    $X' = \{x \in X : \mathcal{L}(x) > v\}$ // $X' \subset R$
    $X_{\text{proj}} = \{Tx : x \in X'\}$
    $S = \text{Boundary}(X_{\text{proj}})$
    return $S$
```

The parameter ranges of the model may have very different scales. To make them visually similar and prevent any dimension from being visually imperceptible, we linearly scale the x, y, and z coordinates prior to calculating the boundary. There are two candidate mappings: rescaling the values onto the unit interval or using the standard score (shift the mean to zero and scale by the standard deviation). Either map makes the scale of all projected dimensions approximately the same. We carry along the original parameter values at each projected point, enabling the viewer to query the original values and ranges.

3.3. Choosing the optimal axis-aligned projection

The surface generated by PercentileRegionProjection is the boundary of the projection into $\mathbb{R}^3$ of the 95th percentile region in $\mathbb{R}^p$. If we limit ourselves to axis-aligned orthonormal projections, there are p-choose-3 possible projections. We let the user select any three axes and display the resulting projection, enabling them to visually explore all three-way interactions among parameters.

Our scientist collaborators expressed particular interest in projection directions that exhibit complex features that cannot be described simply in terms of covariance among sample points. They expect these projections to contain the most scientifically interesting features. Our statistician collaborators refer to distributions with complex features as “banana distributions” because a common such distribution of points in two dimensions resembles a banana. The previously mentioned cup function also exhibits non-simple dependence.

After several incomplete attempts to positively describe what was meant by the term “banana distribution”, we changed our approach and instead found a metric that measures the extent to which a shape is uninteresting – we then look for surfaces that are the least uninteresting. The most uninteresting shape is an ellipsoid, indicating independent parameters, in which case the underlying distributions are well described by their mean value and covariance so that no interesting dependencies between parameters are present. The least interesting ellipsoid is a sphere, where all of the variances match. We also want the metric to locate surfaces that are not homotopic to a sphere because those shapes are certainly interesting. So, rather than seeking “banana” shapes, our metric seeks “non-orange” shapes.

To measure how un-sphere-like a shape is, we measure the membrane energy [18] of the surface. Because this is proportional to the surface area, we normalize by computing the ratio of the square root of the surface area to the cube-root of the volume (the normalized shape index $\mathcal{I}$):

$$\mathcal{I} = \frac{1}{\sqrt[6]{36\pi}} \frac{\sqrt{Area}}{\sqrt[3]{Volume}}$$

Figure 3 shows the value of the normalized shape index for several shapes. Because the surface is generated from a discrete sampling, it tends to have small-scale noise on its surface that is not a result of the underlying distribution. To remove this noise, we apply a smoothing step before calculating the shape index. (We do not include this smoothing in the final visualization, which shows the raw underlying surface.) The scale of the smoothing depends on the user-specified minimum interesting feature size that was used to cull tetrahedra above.

Our selection algorithm projects the points in each of the p-choose-3 different directions and measures the un-sphere-ness of the resulting surfaces. We select the least sphere-like surface and present to the user the three parameters that were used to project the points as well as the surface and the set of projected points. The user can also select any other sets of axes and see the interactions among them to further explore parameter dependencies.

```
OptimalSurfaceProjection $(\mathcal{L}, N, c)$.

$X = \text{Metropolis} - \text{Hastings}(\mathcal{L}, N)$ // $X \subset \mathbb{R}^p$
$v = \text{FindOrderStatistic}(\{\mathcal{L}(x) : x \in X\}, (1-c)N)$
$X' = \{x \in X : \mathcal{L}(x) > v\}$
$T', w' =$ **None**, $-\infty$
**foreach** $T \in$ the $p$-choose-3 axis-aligned projections
    $X_{\text{proj}} = \{Tx : x \in X'\}$
    $X_{\text{proj}} = \text{Rescale}(X_{\text{proj}})$
    $S = \text{Boundary}(X_{\text{proj}})$
    $S = \text{Smooth}(S)$
    $w = \mathcal{I}(S)$
    **if** $w > w'$
        $T', w' = T, w$
$S' = \text{Boundary}(\text{Rescale}(\{T'x : x \in X'\}))$
**return** $S', T'$
```

3.4. Optimal non-axis-aligned projections

It is possible that the parameters chosen by the scientists are not the fundamental parameters of the phenomenon. To search for these more-fundamental axes, and to more concisely represent the parameter space, our collaborators sometimes use principal component analysis to search for combinations that are particularly expressive.

To enable similar searches for the most interesting dependencies between parameters, a variation on our technique rotates the projection direction to consider many randomly chosen non-axis-aligned projections into $\mathbb{R}^3$. The user can specify how many such directions to sample, providing the capability to explore a much broader range of directions than are considered in the original-axis and principal-component directions. This can reveal interesting parameter dependencies in directions that were not previously being investigated by our collaborators.

```
OptimalSurfaceProjection2 $(\mathcal{L}, N, c, n)$.

$X = \text{Metropolis} - \text{Hastings}(\mathcal{L}, N)$ // $X \subset \mathbb{R}^p$
$v = \text{FindOrderStatistic}(\{\mathcal{L}(x) : x \in X\}, (1-c)N)$
$X' = \{x \in X : \mathcal{L}(x) > v\}$
$X' = \text{Rescale}(X')$
$T', w' =$ **None**, $-\infty$
**foreach** $T \in \text{GenerateRandomProjections}(n)$
    $S = \text{Boundary}(\{Tx : x \in X'\})$
    $S = \text{Smooth}(S)$
    $w = \mathcal{I}(S)$
    **if** $w > w'$
        $T', w' = T, w$
$S' = \text{Boundary}(\{T'x : x \in X'\})$
**return** $S', T'$
```

This variation has the advantage that it may find combinations of parameters with interesting relationships. It has the disadvantage that the displayed space is more difficult for the viewer to interpret.

4. Applications

We implemented our approach in an open-source visualization toolkit, inserted it into an open-source application, and tested it on real parameter-space searches from two different science domains.

4.1. Implementation in VTK

We implemented OptimalSurfaceProjection and PercentileRegionProjection as filters for the Visualization Toolkit (VTK) [19]. The first advantage of using this framework is that it is easy to integrate our algorithms into Visualization programs that rely on VTK, such as ParaView [20]. The second advantage is that it makes available implementation of many useful algorithms. We used `vtkThresholdPoints` for the percentile filter, `vtkDelaunay3D` and `vtkDataSetSurfaceFilter` to create a surface from points in space, `vtkSmoothPolyDataFilter` to smooth the surface, and `vtkMassProperties` to calculate the normalized shape index.

We also implemented the algorithm within an extended Visualization Workbench that we developed [21]. This open-source tool extends the ParaView visualization program and makes the technique directly available to our collaborators and to the broader scientific community.

Fig. 4 shows that when run on a single processor for ten thousand points, the optimal-projection code takes less than a minute to select the optimal three dimensions from nine. On our collaborators' 5- and 6-D data sets, it takes less than 15 s. To search PCA coordinates requires another 8–12 s, and up to five additional random projections can be tested every 8–12 s. This search is linearly parallelizable because each projection is independent. This turn-around time enables our collaborators to interactively test different thresholds on their laptops.

Our implementation can be found packaged with our custom version of ParaView, the MADAI Visualization Workbench [21].

4.2. Application to the galaxy formation model

The first science domain tested was a model of galaxy formation [22]. This model starts with an interaction tree that describes how dark-matter particles combined over time to finally form a Milky-Way-like dark-matter halo, the invisible scaffolding of the galaxy. Subsequently, it uses this tree to simulate the time evolution of the baryonic matter that lies within different clumps of dark matter. There are a number of parameters that control the evolution of the baryons in these simulations.

We list the model parameters here by name and function (of meaning to the scientist, just names for the purpose of this paper): Z_r, how massive a dark-matter halo must be soon after the

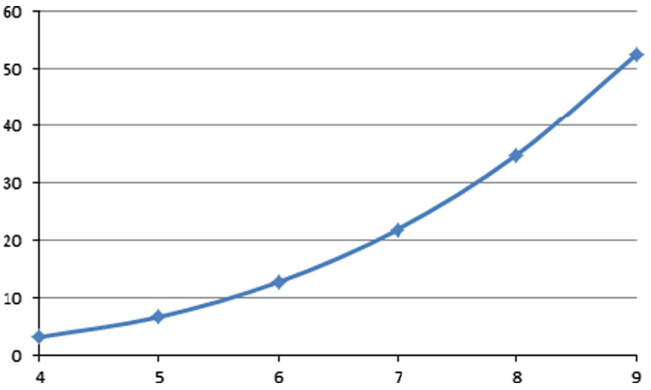

Fig. 4. Seconds taken to compute optimal axes for 10,000 points on a single processor vs. number of axes considered.

big bang to form stars; f_{bary}, the mass fraction of baryons assigned to each dark-matter halo; f_{escp}, the escape factor of metals; *yfe*, the amount of iron released in Super Novae (type II); and *sfe*, the specified star formation efficiency.

When our collaborator looked at the surface created by PercentileRegionProjection, he found it to be more effective than the scatterplot matrix for visualizing and exploring his data. The first important thing he noticed was that, as expected, the optimal axis-aligned projection automatically selected the parameters that show the most complicated interactions. Through use of color (as seen in Fig. 5), he was able to visualize at once relations between four parameters. For this particular problem, exposing and visualizing nonlinear coupling between parameters of the physical processes being considered is of key importance. This is because multiple combinations of different parameters could reproduce equally well the observational data. This is very important as it gives information about what physical process can be better constrained with a given observational data set.

This goal was quickly achieved by our collaborator with optimal projection, avoiding the burden of exploring multiple scatter-plot matrices that only show two parameters at a time. (Scatter-plot exploration also requires mental reconstruction from ambiguous projections.) The projected 95 percentile surface tightly encloses the region where the scientists' model is likely to reproduce reasonably well the observational data. Thus, the observed interactions between parameters were meaningful to him. For example, it became instantly very clear that, while parameters Z_r and f_{bary} are strongly non-linearly coupled, parameters f_{bary} and *sfe* have a more linear relationship. A second important feature reported by the scientist was the ease with which he was able to explore more restrictive iso-likelihood optimally projected surfaces. Simply by modifying the value of the percentile for the region selection, he could quickly explore whether the previously observed couplings were preserved as he selected more restrictive cuts. He reports that the insight he gained using our method is not easily achieved by looking at the 2D scatter plots. The regions of interest and observed couplings showed the scientist where to further explore the data by running the models, thus probing them more closely.

4.3. Application to the RHIC collision model

The second science domain model tested was a simulation of collisions between sets of gold nuclei at the Relativistic Heavy Ion Collider (RHIC) [23]. The version of the model we tested has six parameters.

The parameters of this model (again, just names for the purpose of this technique but with meaning to our collaborators) are the following: $(dE/dy)_{pp}$, the initial energy per rapidity in the diffuse limit compared to measured value in pp collision; σ_{sat}, which controls how saturation sets in as a function of areal density of the target or projectile; f_{wn}, the relative weight of the wounded-nucleon and saturation formulas for the initial energy density; F_{flow}, the strength of the initial flow; $\eta/s|T_c$, the viscosity to entropy ratio for a temperature $T=170$ MeV; and α, the temperature dependence of η/s for temperatures above 170 MeV/c.

When the RHIC model's likelihood density function was plotted using a scatterplot matrix, it revealed no interesting pairwise relationships that lie along the initial parameter axes. When viewing these, our collaborators were left with a nagging doubt that perhaps there was an undiscovered dependency lying along some other projection direction. They tried to address this by running principal-component analysis on the parameters and then viewing the pairwise projections in those spaces. In this way, they sampled two sets of linear combinations of the axes to try and discover hidden features.

Optimal Percentile Region Projection also showed a relatively compact three-dimensional shape without interesting features. Because it had sampled a large space of potential axis combinations (the user can let the algorithm run as long as they like), because it displays relationships between three axes, and because it directly shows the most-interesting projection direction of all those that have been found, it provided more compelling evidence that the parameter space is well-explained by the statistical correlation values.

Even when Optimal Percentile Region Projection reveals that there are no interesting three-dimensional features, Percentile Region Projection can still be used along with a scalar color map to visualize the relationship between four parameters at a time (see Fig. 6), or between three parameters and a model output scalar field. Our collaborators found that these visualizations rapidly express information about these relationships that two-dimensional scatterplots do not.

5. Conclusion

It is difficult to understand the potentially complex relationships among parameters in scientific models when there are many parameters. Optimal Percentile Region Projection makes salient features of the high-likelihood regions of parameter space that cannot be seen using other methods and more clearly shows that there are no interesting three-way relationships when there are not.

By displaying the projection of only the region of high likelihood in the high-dimensional space, rather than the region of high likelihood in the lower-dimensional projected space, we directly address statistical questions about the original parameter space. By choosing the projection that is most different from simple correlation, we save scientists the time and frustration of searching all *p*-choose-3 possible projections.

This technique extends the display of parameter dependencies from the standard two dimensions up to three geometric dimensions, with an overlaid fourth dimension (input or output) shown using color. It uses orthographic projection, which avoids adding perspective or other more complicated distortions in the projection step.

6. Limitations and future work

As with all projection techniques, the presented work hides information. Future work will be needed to address features that cannot be visualized in three dimensions. The addition of layered surface textures or glyphs may be able to extend beyond the four dimensions shown here.

The choice of displaying the boundary of the projected region as an opaque surface hides any interior holes in the region when shown in 3D. Revealing these interior voids will require the addition of cutaway slices or other techniques to display nested surfaces [24–26].

Our method is targeted at likelihood density functions that have a single region of high value. We focused on this case because the real-world examples from our scientists behaved in this manner. An example of a method that can extract topological information about multiple high-value regions of a scalar function is Topological Spines [11]. Future work could combine these techniques, using Topological spines to show the distribution of local maxima and our projection technique to display each local region.

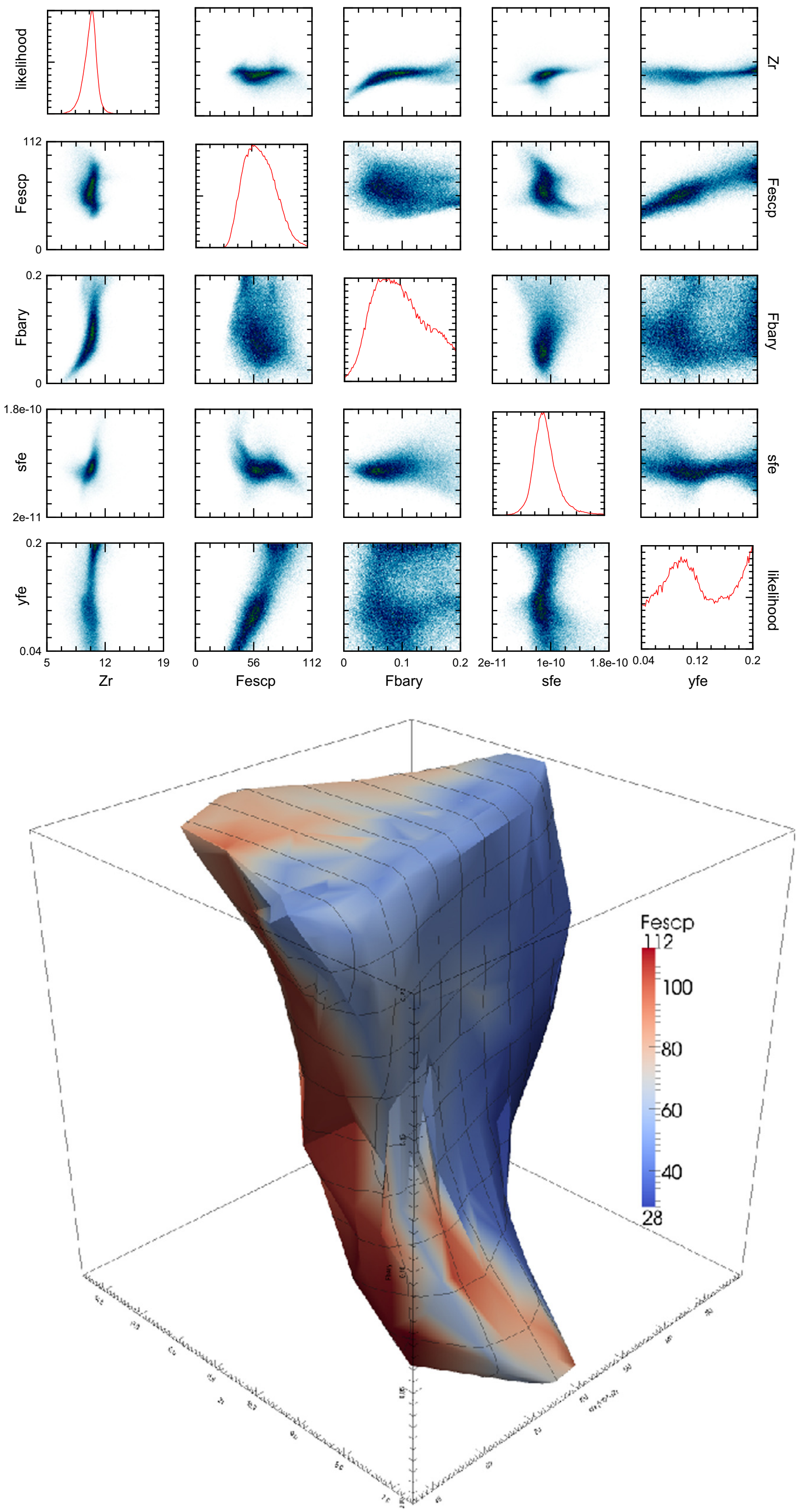

Fig. 5. A comparison of the scatterplot matrix and Optimal Percentile Region Projection of the galaxy formation model likelihood. With Optimal Percentile Region Projection, we visualize the complex relationship among all four parameters. The algorithm chose to project into the Z_r, f_{bary}, sfe space, and we then colored the surface by f_{escp}.

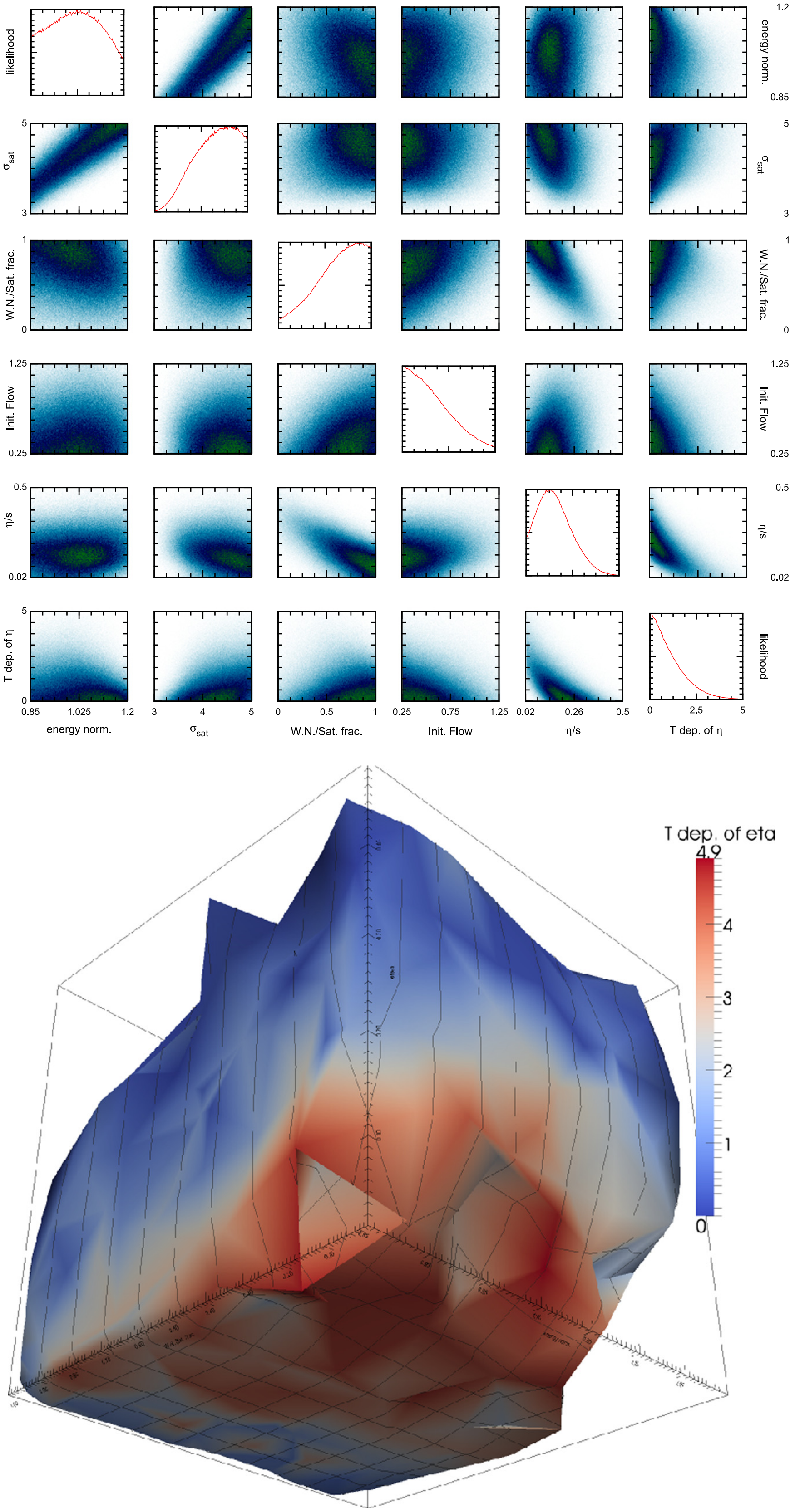

Fig. 6. A comparison of the scatterplot matrix and Optimal Percentile Region Projection of the RHIC model likelihood. The Optimal Percentile Region Projection chose to project into the $(dE/dy)_{pp}, f_{wn}, \eta/s$ space. Both the scatterplot matrix and the projections reveal that there are no complex relationships among model parameters, however the Optimal Percentile Region Projection is able to search a much-larger space of linear combinations of axes to locate unexpected relationships so it provides more compelling evidence of a lack of dependencies.

Acknowledgments

This work was supported by the U.S. National Science Foundations Cyber-Enabled Discovery and Innovation Program through Grant no. 0941373 and by Sandia National Laboratories contract number 979573.

References

[1] Gelfand AE, Smith AF. Sampling-based approaches to calculating marginal densities. J Am Stat Assoc 1990;85(410):398–409.

[2] Engel D, Huttenberger L, Hamann B. A survey of dimension reduction methods for high-dimensional data analysis and visualization. In: Garth C, Middel A, Hagen H, editors. VLUDS; OASICS, vol. 27. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, Germany. ISBN 978-3-939897-46-0; 2011. p. 135–49. URL 〈http://dblp.uni-trier.de/db/conf/vluds/vluds2011.html#EngelHH11〉.

[3] van der Maaten LJP, Postma EO, van den Herik HJ, Dimensionality reduction: a comparative review, Journal of Machine Learning Research - JMLR 01/2007 〈http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.125.6716&rep=rep1&type=pdf〉.

[4] Williams M, Munzner T. Steerable, progressive multidimensional scaling. In: INFOVIS: IEEE symposium on information visualization; 2004. p. 57–64. http://dx.doi.org/10.1109/INFVIS.2004.60.

[5] Ingram S, Munzner T, Irvine V, Tory M, Bergner S, Moller T. Dimstiller: workflows for dimensional analysis and reduction. In: VAST: IEEE symposium on visual analytics science and technology; 2010. p. 3–10. http://dx.doi.org/10.1109/VAST.2010.5652392.

[6] Fodor I. A survey of dimension reduction techniques. Technical Report; 2002.

[7] David Feng Yueh Lee LKRT. Matching visual saliency to confidence in plots of uncertain data. IEEE Trans Vis Comput Graph 2010;16(6):980–9.

[8] Jonathan M Harter, Russell M Taylor II, X.W.C.H.S.B.S.Z. Increasing the perceptual salience of relationships in parallel coordinate plots. In: Proceedings of the SPIE visualization and data analysis; 2012. p. T1–12.

[9] Inselberg A. Parallel coordinates: visual multidimensional geometry and its applications. New York, USA: Springer; 2009.

[10] Oesterling P, Heine C, Weber GH, Scheuermann G. Visualizing nd point clouds as topological landscape profiles to guide local data analysis. IEEE Trans Vis Comput Graph 2013;19(3):514–26.

[11] Correa C, Lindstrom P, Bremer PT. Topological spines: a structure-preserving visual representation of scalar fields. IEEE Trans Vis Comput Graph 2011;17(12):1842–51.

[12] Buja A, Cook D, Swayne DF. Interactive high-dimensional data visualization. J Comput Graph Stat 1996;5:78–99.

[13] Borgo R, Kehrer J, Chung DH, Maguire E, Laramee RS, Hauser H, et al. Glyph-based visualization: foundations, design guidelines, techniques and applications. In: Eurographics State of the Art Reports. EG STARs. Eurographics Association; 2013. p. 39–63. 〈Http://diglib.eg.org/EG/DL/conf/EG2013/stars/039-063.pdf〉; URL 〈http://www.cg.tuwien.ac.at/research/publications/2013/borgo-2013-gly/〉.

[14] Healey CG, Enns JT. Attention and visual memory in visualization and computer graphics. IEEE Trans Vis Comput Graph 2012;18(7):1170–88.

[15] Huber DE, Healey CG. Visualizing data with motion. In: VIS: IEEE visualization conference; 2005. p. 527–34.

[16] Hastings WK. Monte carlo sampling methods using Markov chains and their applications. Biometrika 1970;57(1):97–109.

[17] Press WH. Numerical recipes: the art of scientific computing. 3rd ed. Cambridge, UK: Cambridge University Press; 2007.

[18] Crane KM. Conformal geometry processing [Ph.D. thesis], California Institute of Technology; 2013.

[19] Schroeder W, Lorenson B. Visualization toolkit: an object-oriented approach to 3-D graphics. 1st ed. Upper Saddle River, NJ, USA: Prentice Hall PTR; 1996 ISBN 0131998374.

[20] Squillacote A. The ParaView Guide: a parallel visualization application. Kitware; 2007. ISBN 9781930934214. URL 〈http://www.kitware.com/products/books/paraview.html〉.

[21] The MADAI Collaboration MADAI Visualization Workbench. URL 〈http://vis.madai.us/〉; 2013.

[22] Gómez FA, Coleman-Smith CE, O'Shea BW, Tumlinson J, Wolpert RL. Characterizing the formation history of milky way like stellar halos with model emulators. Astrophys J 2012;760(2):112.

[23] Novak J, Novak K, Pratt S, Vredevoogd J, Coleman-Smith C, Wolpert R. Determining fundamental properties of matter created in ultrarelativistic heavy-ion collisions;2013. arXiv:13035769[nucl-th] 〈http://arxiv.org/abs/1303.5769〉.

[24] Interrante V, Fuchs H, Pizer S. Illustrating transparent surfaces with curvature-directed strokes. In: VIS: IEEE visualization conference; 1996. p. 211–18.

[25] Weigle C, II, RMT. Visualizing intersecting surfaces with nested-surface techniques. In: VIS: IEEE visualization conference; 2005. p. 503–10.

[26] Alabi OS, Wu X, Bass S, Pratt S, Zhong S, Healey C, et al. Exploring ensemble data sets through ensemble surface slicing. Proc SPIE Vis Data Anal 2012;8294:U1–12.