Visualizing Static Ensembles For Effective Shape and Data Comparison

Lihua Hao, Christopher G. Healey, Steffen A. Bass, and Hsuan-Ya Yu;
North Carolina State University and Duke University; Raleigh, North Carolina

Abstract

Ensembles are large, multidimensional, multivariate datasets generated in areas like physical and natural science to study real-world phenomena. Simulations or experiments are run repeatedly with slightly different initial parameters, producing members of the ensemble. The need to compare data and spatial properties, both within an individual member and across multiple members, makes analysis challenging. Initial visualization techniques focused on ensembles with a limited number of members. Others generated overviews of larger ensembles, but at the expense of aggregating potentially important details. We propose an approach that combines these two directions by automatically clustering members in ways that help scientists locate interesting subsets, then visualize members within the subset. Our ensemble visualization technique includes: (1) octree comparison and clustering to generate a hierarchical level-of-detail overview of inter-member shape and data similarity; (2) a glyph-based visualization of an ensemble member; and (3) a method of combining multiple glyph visualizations to highlight similarities and differences in shape and data values across a subset of ensemble members. We apply our approach to a Relativistic Heavy Ion Collider ensemble collected by nuclear physics colleagues at Duke University studying quantum chromo-dynamics. Our system allows the physicists to interactively choose when to explore inter-member relationships, and when to visualize fine-grained details in individual member datasets.

Introduction

An *ensemble* is formed by executing a simulation or an experiment repeatedly, with slightly different initial conditions or parameterizations for each run. Data produced from a run forms one *member* of the ensemble. Researchers from a wide range of disciplines are now using ensembles to investigate complex systems, explore a system's sensitivity to its input parameters, measure uncertainty, and compare both spatial and data characteristics of the resulting models.

Not surprisingly, ensembles are difficult to analyze due to their size and complexity. Wilson et. al. compared ensembles to traditional scientific data and summarized the characteristics and challenges unique to ensemble visualization [25]. Different techniques have been developed for ensemble analysis. One approach creates concise overview visualizations, but these may hide potentially important details in the original data [3, 20]. Another method extends existing scientific visualization techniques to support comparison between members [1, 17]. This can offer an improved view of individual members, but often cannot scale beyond small member sets. This suggests the two main approaches to ensemble visualization are currently: (1) generate an overview that scales but may not maintain detail, or (2) present a visualization that maintains detail but can only analyze a small number of members at one time. More recent systems try to support interactive ensemble analysis at different levels of detail [12, 18]. These systems rely on the scientists to select a subset of members for detailed visualization, however. Currently, little work has investigated ways to automatically capture inter-member relationships.

We propose an approach that combines the two directions of ensemble analysis. A key strength of our method is the automatic construction of hierarchical representations of ensembles based on their shape and data similarity. The hierarchy is visualized to the scientists, allowing them to use their current interests and domain expertise to control the trade-off between individual member detail versus the number of members being visualized. Our technique reveals hierarchical inter-member relationships and supports visualization of both a single member and multiple member subsets.

We use an octree representation to compress the data and extract shapes from the ensemble [9, 21]. The hierarchical structure of the octree naturally encodes shapes and variations between members at multiple levels of detail. We extend the similarity matching in [26] to mathematically measure shape dissimilarity between member pairs by comparing their octrees. Based on these estimates, we apply hierarchical clustering to collect similar members into common groups. The result is a level-of-detail cluster tree visualization that allows scientists choose where to perform comparative analysis by interactively selecting individual member datasets or clusters of members with varying levels of similarity.

Next, we represent member and inter-member relationships with a visualization technique that displays the members within a cluster. We merge member data using statistical aggregation into a visual presentation that highlights shape and data differences through the use of size, colour, and motion. In this way, we extend traditional multivariate visualization to support general shape visualization and region-by-region comparative visualization across multiple ensemble members. This provides a detailed view of shape, data element distributions, and important attribute value differences across the members in a cluster.

Related Work

In the past decade, different visualization techniques have been proposed to facilitate interpretation and analysis of 2D or 3D ensemble data using volume rendering, multidimensional visualization, and comparative visualization [2, 10, 16].

Noodles is a visualization technique designed to analyze meteorological ensembles [22]. It includes statistical aggregation and uncertainty measurements, visualizing results with circular glyphs, ribbons, and *spaghetti plots*, a visualization method that uses contours to represent attribute value boundaries. Ensemble-Vis also focuses on statistical data visualization for analyzing weather forecast and climate model ensembles [19]. Ensemble-Vis presents data using a collection of visualizations connected through linked views. Data from multiple member sets are summarized with means and standard deviations, then visualized using colour maps, contours, height fields, trend charts, and spaghetti plots.

Follow-on research extends ensemble visualization to explicitly support member comparison. Ensemble Surface Slicing (ESS) compares surfaces extracted from *n* ensemble members in a single view by colour-coding the members, then slicing them into equal-width strips [1]. A combined representation is built by abutting strips member-by-member, where every *n*-th strip belongs to a common member, and visual discontinuities between strips highlight surface shape differences. Phadke et. al. proposed: (1) pairwise sequential animation, and (2) screen door tinting for 3D ensemble visualization [17]. Pairwise sequential animation extracts data elements from a member, visualized as glyphs whose colour and shape represent attribute value and parent member, respectively. Screen door tinting divides a projected ensemble visualization into equal sized cells whose colour and luminance identify a cell's parent member and differences versus a user defined reference member, respectively.

Recently, Matkovic et. al. developed a visualization tool to interactively investigate ensembles as families of 2D data surfaces. [12]. The system presents projections and aggregations of the data surfaces at three different levels: a parallel coordinate and scatterplot level to explore correlations and trends in data attributes; a parallel coordinates level to explore relationships across surfaces through aggregated profiles and function graphs; and 2.5D or 3D height fields to to support in-depth analysis of a selected surface. Piringer et. al. designed a system for comparative visual analysis of 2D function ensembles [18] using: (1) a domain-oriented overview that aggregates features across an ensemble using a heatmap; (2) a member-oriented overview that visualizes members as icons in a scatterplot; and (3) a detailed member view that presents small subsets of members in a 3D scatterplot.

Whitiker and Mirzargar developed specialized contour and curve boxplots to accurately visualize statistical properties, outliers, and variability in ensembles of contours or 2D and 3D curves [13, 24]. They statistically summarize the centrality of members in an ensemble, visualized using specialized boxplots. Demir developed a method of overlaying bar and line charts to present statistical summaries and variations in ensemble members [4]. Köthur focused on temporal aspects of ensembles, generating clusters from temporal profiles of different members to support feature identification and ensemble comparison [11].

Past research shows numerous examples of ensemble visualization research built on previous techniques like glyphs, comparative visualization, charts, and linked views. We adopt a similar approach in our work, which is perhaps most similar to the contour and curve boxplots of Whitaker and Mirzargar [13, 24]. Their goals differ from ours, however. Contour boxplots visualize contours and functional level sets within an ensemble. We are focused on defining a hierarchical representation of 3D ensemble members that support both shape and value comparison across multiple members.

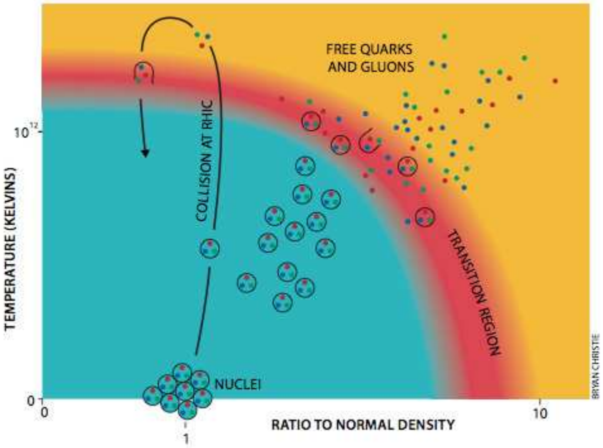


Figure 1: A calculated transition from ordinary nuclei to free quarks and gluons, where protons and neutrons within the nuclei disintegrate at extremely high temperature or density

To achieve this goal, we focus on two critical issues in ensemble visualization: (1) scalability to larger member sets; and (2) visualizations that allows scientists to make informed decisions about how to trade-off individual member detail against the number of members being compared. We measure shape dissimilarities between ensemble members, hierarchically combining members with similar shapes into clusters for more detailed exploration. Clustering uses an octree-based ensemble visualization framework that offers: (1) a mathematical measure of shape similarity between 3D spatial ensemble members; (2) a cluster tree visualization that provides a level-of-detail hierarchical overview of inter-member relationships prior to the need for detailed comparisons; (3) more concise visual representations for multiple members to improve scalability; and (4) glyph-based visualization of a single member or multi-member subsets that highlight similarities and differences in both shape and attribute value.

RHIC Ensemble

We are collaborating with nuclear physicists from Duke University to study quark–gluon formation from the Relativistic Heavy Ion Collider (RHIC) at Brookhaven National Laboratory[1]. Heavy ion collisions at very high energies are used by physicists to investigate interacting matter under extreme conditions[15]. Real-world and simulation results are used to estimate quantum chromo-dynamics (QCD), a quantum field theory of strong interactions. Calculations confirm QCD matter transition from hadronic gas to quark–gluon plasma (QGP) occurs at extremely high temperature and energy densities. In the QGP phase, protons and neutrons in the nuclei break up, releasing quarks and gluons (Figure 1).

Interest in quark–gluon plasma revolves around the belief that this energy existing in the universe during the first few microseconds of the Big Bang. The RHIC allows our scientists to collide two opposing gold nuclei head-on at relativistic speeds [14]. These collisions produce very hot, very dense bursts of matter and energy that simulate conditions in the very early universe during the QGP phase. This is often termed "the little bang in the

[1] `www.bnl.gov/rhic/`

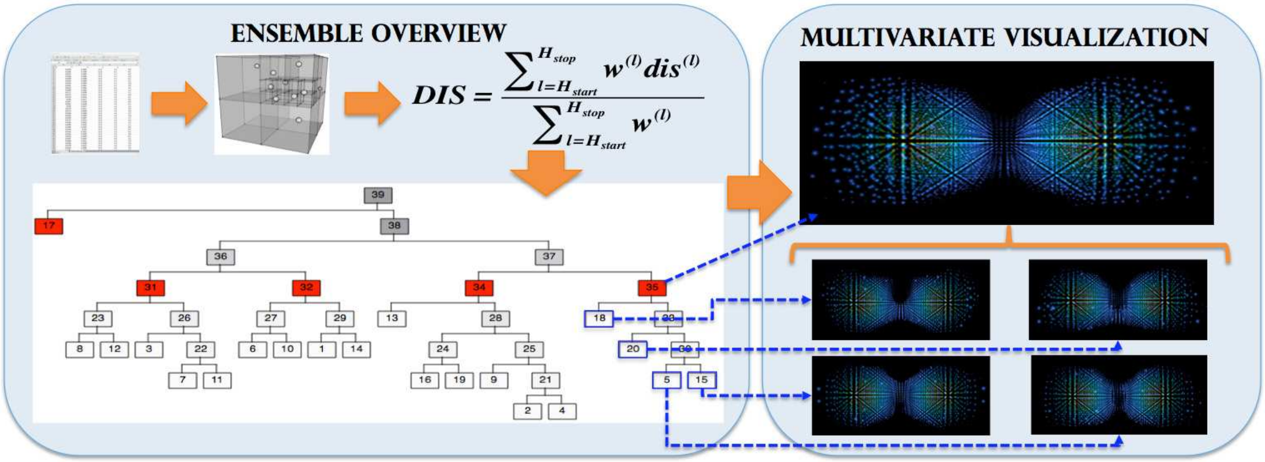

ENSEMBLE OVERVIEW
$DIS = \frac{\sum_{l=H_{start}}^{H_{stop}} w^{(l)} dis^{(l)}}{\sum_{l=H_{start}}^{H_{stop}} w^{(l)}}$
MULTIVARIATE VISUALIZATION
39
17
38
36
37
31
32
34
35
23
26
27
29
13
28
18
33
8
12
3
22
6
10
1
14
24
25
20
30
7
11
16
19
9
21
5
15
2
4

Figure 2: A system diagram showing the cluster tree overview and the multivariate member visualizations

laboratory.” A key requirement is assigning proper initial conditions and input parameters for the collisions. The results from different inputs are compared to identify similarities and differences in hydrodynamic evolution. One main goal during the analysis is to identify the critical point where the QGP state transitions to a hadronic final state.

Availability limits the number of RHIC experiments the physicists can perform. Because of this, results from real-world RHIC collisions are used to build models that simulate hydrodynamic evolution. Based on a hydrodynamic calculations of a gold on gold collision, the full ensemble contains hundreds of members from simulation runs with varying: (1) quantum fluctuations of protons and neutrons; (2) start times for the hydrodynamic calculations; and (3) granularities of the initial energy–density deposit that enters the hydro field. Each member contains a large number of 3D spatial data elements with the attributes: (1) temperature; (2) energy density; (3) net baryon density (baryons are particles made up of three quarks); (4) baryo-chemical potential; (5) pressure; (6) fraction of quark–gluon plasma; and (7) velocity.

Our physicists are interested in how varying the initial parameters affects the evolution of shape and data dissimilarities throughout the simulation. Differences among RHIC simulations (*i.e.*, ensemble members) contain important information related to these asymmetries. Our hierarchical clustering ensemble visualization allows the physicists to look at different levels of ensemble aggregation and isolate subclasses of simulations that may contain interesting or unique information.

Design

We define an ensemble $E = \{m_1, m_2, \dots m_N\}$ with N members $m_i \in E$. Our system analyzes and visualizes E at two levels: (1) as an overview of shape similarity-based inter-member relationships, visualized as a cluster tree; and (2) as a glyph visualization for detailed exploration and comparison of ensemble members (Figure 2).

We begin with an octree construction that extracts ensemble member shapes at different levels of detail. We measure shape differences between all pairs of members by comparing their octree representations, then apply hierarchical clustering to build a cluster tree visualization that reveals inter-member relationships. This is done prior to performing detailed visual comparisons, allowing for rapid overview construction. Scientists interact with the cluster tree to determine which subsets of members to examine in detail. Each subset is visualized using a multivariate 3D glyph visualization that highlights similarities and differences between members. We discuss these techniques, including a description of the original approaches and how we extended them to support ensemble analysis and visualization.

3D Shape Octrees

Octrees are widely used for memory reduction in 3D model storage. An octree is a 3D analogy of a quadtree, where each node is recursively subdivided into eight children [9, 21]. Subdivision terminates when a stopping condition is reached, for example, when the height of the octree reaches a user defined maximum level.

We use a single octree to extract the shapes—or more specifically, the spatial distribution of a member’s data elements—for all N members in the ensemble. The octree is then used to perform shape-based member comparison and aggregation. Traditionally, an octree is built for a single 3D model. To support inter-member shape comparison, we construct an octree that encodes data for multiple members. Octree construction begins with a root node representing the minimum bounding cube that covers all the data elements in E. For each member m_i we recursively subdivide the root octant into eight equal-sized, non-intersecting child octants until the number of elements within an octant is less than or equal to a user defined upper bound $P_{\max}$ or the height of octree reaches a user defined maximum depth $H_{\max}$. To save memory and reduce compute time, we do not create empty octants that contain no data elements from any m_i.

Once construction is complete, each octant contains data from q members, $1 \le q \le N$. Data from each member m_i is aggregated to encode the following information in an octant: (1) q summarized data points representing the average spatial location of each m_i’s data elements; and (2) q average–variance pairs (μ_i, σ_i) representing the average and variance of the attribute values stored in each m_i’s data elements.

Several features of the octree inspire us to use it in ensemble analysis. An ensemble member normally contains a large number of unorganized data elements. For example, RHIC members contain between 180,000 and 3,300,000 data elements. This makes them expensive to store and render, especially when rotation, translation, or animation are involved. Some method to reduce the size of the data is needed. Existing ensemble visualization algorithms (*e.g.*, pairwise sequential animation [17]) use clustering algorithms to select a subset of data elements that match a spatial distribution of attribute values, but they do not correlate elements from different members and cannot easily perform similarity calculations. An octree representation not only reduces data size by aggregating data elements in an octant, but it also links spatially related elements from different members by assigning them to a common octant. This enables octant-by-octant shape comparison. Additionally, octrees naturally extract 3D shapes at multiple levels of detail, adding flexibility to the resulting visualization and shape comparison. For instance, a RHIC member with 712,740 data elements represented by an octree with $P_{\max} = 300$ contains 5,872 octants.

Shape Dissimilarity

Previous ensemble visualizations rely on humans to intuitively measure differences or correlations between ensemble members. We provide a mathematical measure of pairwise member shape dissimilarity based on the members' octree representations. We define the *shape* of a member as the distribution of its data elements in 3D space, and not simply its outer surface position. Our shape dissimilarity measure lays a foundation for hierarchical overviews of inter-member relationships. Scientists do not have to predict relationships between members ahead of time to decide which subset of members to analyze and visualize.

Our member shape comparison algorithm is inspired by Zhang and Smith's work on octree shape similarity matching for 3D shape retrieval [26]. Where they built octrees independently for each object, we generate a single, consistent octree representation whose root covers data elements from all members of E. With independent octrees, data elements for a member may distribute in a small subregion of the root octant. In this case, two members with significantly different shapes will be incorrectly assigned a high similarity because they have numerous empty octants in common. This is one reason why we exclude empty octants from our octree. Additionally, to ensure an upper similarity bound $sim_{i,j}$ of 1 between members m_i and m_j, each octant in Zhang's algorithm always contains eight children. This may not be true in our octree, so we adjust the similarity algorithm to maintain this upper bound guarantee.

To support follow-on shape clustering, we measure dissimilarity between members, as opposed to similarity. We modify Zhang's algorithm to maintain dissimilarity accuracy for octrees with large common empty regions. To compare the shapes of m_i and $m_j \in E$, we calculate $dis^r_{i,j}$, the dissimilarity score between m_i and m_j in the r-th octant o^l_r at level l in the octree. cnt^r_i and cnt^r_j represent the number of data elements of m_i and m_j that lie within o^l_r. The calculation ignores any octant that is empty for both members. It considers m_i and m_j as equivalent at o^l_r if $cnt^r_i = cnt^r_j$ ($dis^r_{i,j} = 0$), as completely different if either cnt^r_i or cnt^r_j is 0 ($dis^r_{i,j} = 1$), and as partially different otherwise, measured as:

$$dis^r_{i,j} = \frac{\left|cnt^r_i - cnt^r_j\right|}{\max\left(cnt^r_i, cnt^r_j\right)} \tag{1}$$

dis^l_r ranges from 0 to 1, with higher scores representing larger relative differences in point counts between m_i and m_j.

Given Eq. 1 for a single octant, we must we aggregate dissimilarities between m_i and m_j across all octants in the octree. For octree level l with N^l non-empty octants, the dissimilarity between m_i and m_j is:

$$dis^l = \frac{\sum_{r=1}^{N^l} dis^r_{i,j}}{N^l} \tag{2}$$

Since the maximum value of $dis^r_{i,j}$ is 1 (Eq. 1), N^l is the maximum value for $\sum_{r=1}^{N^l} dis^r_{i,j}$, producing $0 \le dis^l \le 1$.

Finally, we aggregate dissimilarities over all levels, starting at the root, to create an overall dissimilarity score. Given octree height H, the final dissimilarity score $dis_{i,j}$ between m_i and m_i is:

$$dis_{i,j} = \frac{\sum_{l=1}^{H} w^l dis^l}{\sum_{l=1}^{H} w^l} \tag{3}$$

$w^l = 1/\gamma^l$ is used to weight the dissimilarities at different levels in the octree according to a shape comparison factor γ. If $0 < \gamma < 1$, larger weights are assigned to more detailed octree levels (*i.e.*, levels farther from the root). If $\gamma > 1$, larger weights are assigned to more abstract levels (*i.e.*, levels closer to the root). Setting $\gamma = 1$ weights all levels equally. The range of $dis_{i,j}$ is $[0,1]$ where $dis_{i,j} = 0$ indicates full similarity and $dis_{i,j} = 1$ indicates complete dissimilarity.

In practice, we may not always want to compare the octree at all levels. A point number comparison at the root is probably too abstract and a shape comparison at the leaves may be too detailed. To provide more flexibility, we allow the dissimilarity calculation to start at a user-specified level H_{start} and stop at level H_{stop}, $1 \le H_{start} \le H_{stop} \le H$, so that abstract shape information above H_{start} and detailed shape information below H_{stop} will be ignored.

Our octree comparison counts the number of data elements for each member in an octant and measures their relative differences. Higher dissimilarity scores imply a higher percentage of differences. Multiple levels of shape detail are considered, and comparisons can be focused on more abstract or more detailed levels in the octree with starting and stopping levels, and weights based on the shape comparison ratio γ. This flexibility allows scientists to adjust the shape measurement strategies to fit to their interests.

Cluster Trees

The shape dissimilarity calculations produce an $N \times N$ dissimilarity matrix encoding shape differences between all member pairs. We use the dissimilarity matrix to perform hierarchical clustering, organizing members into groups with similar shapes. Cluster results are visualized as a cluster tree to provide scientists with a better understanding of inter-member shape relationships in the ensemble.

We initially implemented two clustering techniques: *minimum spanning tree (MST) clustering*, a top-down hierarchical clustering procedure, and *agglomerative clustering*, a bottom-up hierarchical clustering procedure. MST clustering is intuitive, easy to implement, and works well on a variety of datasets, particularly when clusters do not exhibit spherical shapes. Agglomerative clustering iteratively merges the two most similar clusters and

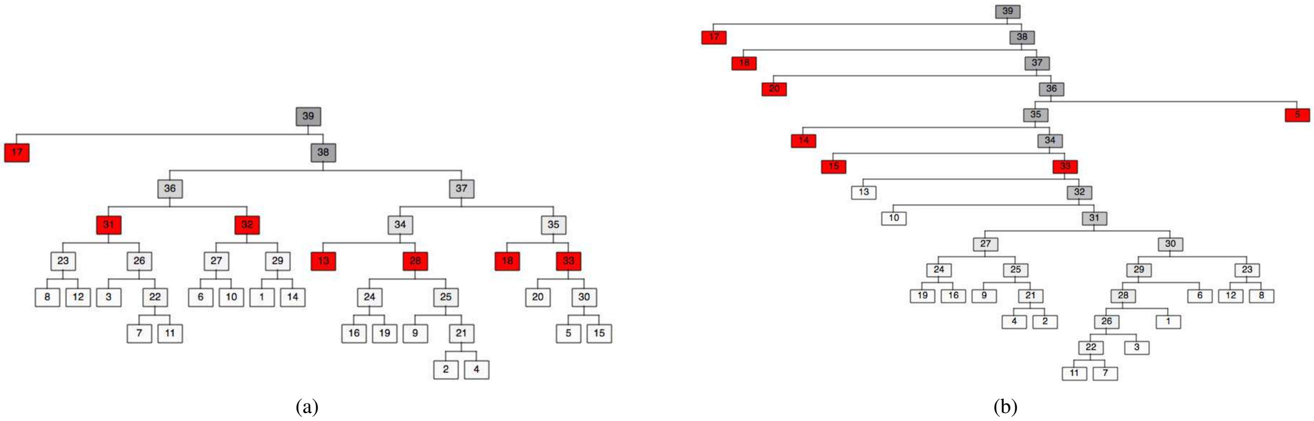


Figure 3: Cluster trees, red nodes highlight clusters for $k = 7$: (a) agglomerative clustering results for a 20-member RHIC ensemble; (b) MST clustering results for the same 20-member RHIC ensemble

updates the dissimilarity matrix until the members are assigned to k clusters, or the dissimilarity between the two most similar clusters falls below a predefined threshold τ.

Our comparison of the two techniques showed that agglomerative clustering results are often better than MST results, since agglomerative clustering updates dissimilarity between clusters at each iteration. This produces more balanced cluster trees (agglomerative cluster tree, Figure 3a versus MST cluster tree, Figure 3b). Because of this, we focused on agglomerative clustering, although the option of MST clustering is still available to the user.

A key procedure in agglomerative clustering is updating the dissimilarity matrix when two clusters are merged, to measure the dissimilarity between new and existing clusters. Let m_i and m_j be any members in clusters S and T, $dis_{i,j}$ be the dissimilarity between m_i and m_j, $dis_{S,T}$ be the overall dissimilarity between clusters S and T, and $|S|$ and $|T|$ be the number of members in S and T. We considered three different methods to measure dissimilarity between clusters:

1. *Complete-linkage* chooses the maximum dissimilarity between all possible member pairs: $dis_{S,T} = \max(dis_{i,j}) \forall i \in S, j \in T$.
2. *Single-linkage* chooses the minimum dissimilarity between all possible member pairs: $dis_{S,T} = \min(dis_{i,j}) \forall i \in S, j \in T$.
3. *Group average linkage* calculates the mean dissimilarity between all member pairs: $dis_{S,T} = \frac{1}{|S| \cdot |T|} \sum_{i \in S} \sum_{j \in T} dis_{i,j}$.

In practice, group average linkage normally provides better cluster dissimilarities than the other two methods, but at the cost of a more expensive calculation.

Applying agglomerative clustering until all members belong to a single cluster produces a series of clustering results that assign members into $k = N$, $k = N - 1$, …, $k = 1$ clusters. Figure 3a shows the agglomerative cluster tree visualization of a 20-member RHIC ensemble. The red nodes highlight the clustering result defined by $k = 7$.

The resulting cluster tree visualization (Figure 3a) provides a hierarchical level-of-detail overview of inter-member relationships, making it easier for scientists to choose a subset of members to compare, analyze, and visualize.

User Interaction

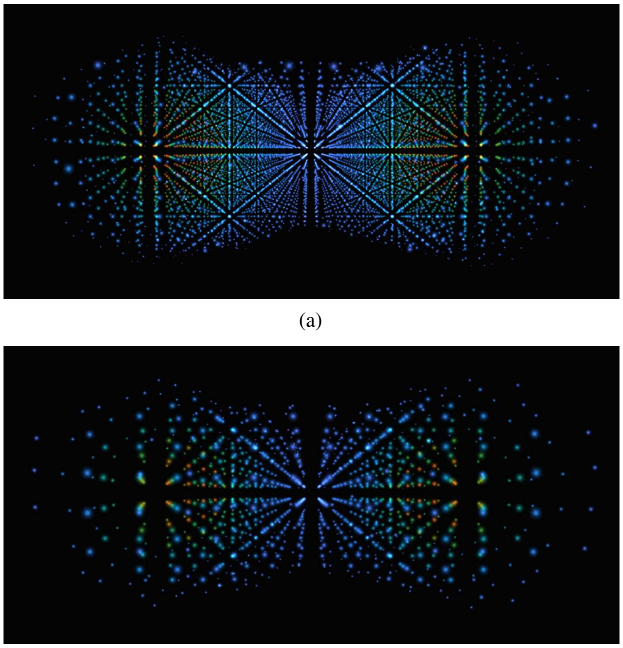
Our system initially presents the cluster tree to allow users to interactively choose which sets of members to explore. Selecting nodes higher in the tree presents an overview of numerous members, while selecting nodes lower in the tree visualizes similarities and differences between only a few members. This allows users to trade off the number of members being visualized versus presenting details for individual members. More importantly, it allows users to apply their domain expertise and knowledge of context to choose appropriate member sets. In this way, the cluster tree forms a hierarchy that allows users to visualize members at the desired level-of-detail as their investigations unfold.

Once a member or a set of members is selected, their shape and attribute values are visualized with a glyph-based technique. The volume can be manipulated in the standard ways: translation to move around and through the volume, rotation to view the volume from different perspectives, and zoom to focus on subsets of interest within the volume.

Ensemble Member Visualization

We designed two glyph-based visualizations to display 3D ensemble members represented by octrees: a single member visualization and a cluster visualization. The single member visualization displays detailed distributions of shape and attribute value for one ensemble member. The cluster visualization displays a summarization and comparison of shapes and attribute value distributions for multiple members.

The basic foundation for both visualizations was inspired by previous work on perceptual and nonphotorealistic visualization techniques [7, 23]. Low-level cognitive vision occurs in two stages: *orientation*, where the visual system chooses to orient to a particular location in an image, and *engagement*, when the visual system may choose to linger and obtain visual details at that location. Although we know how to orient a viewer, understanding what causes the visual system to engage is still an open problem. One hypothesis we have been studying is that the perception of aesthetic beauty may promote engagement. Preliminary results have been promising, suggesting that the increased cost of creating nonphotorealistic visualizations may be justified by an increased memory for detail versus a more traditional representation.



(a)

(b)

Figure 4: Single member visualizations: (a) all leaf octants visualized; (b) an abstract visualization with $H_{\max} = 5$ in a six-level octree

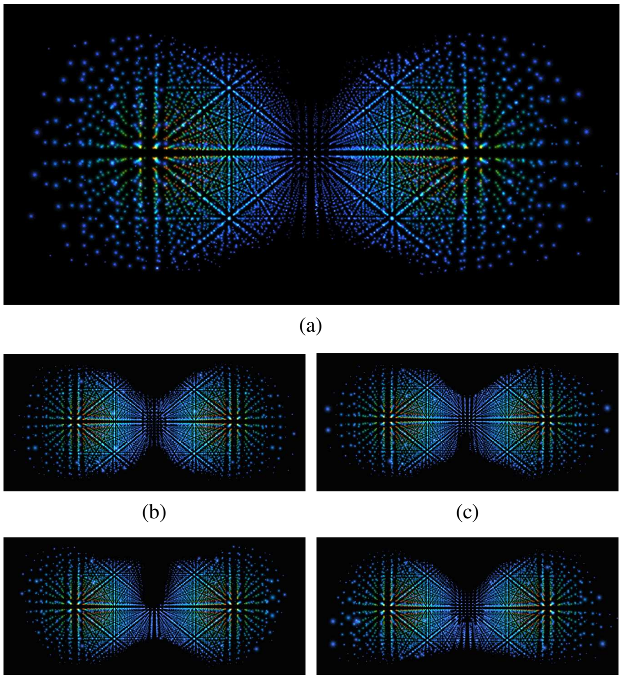
Our ensemble visualizations were initially designed to mimic star field patterns, similar to what you might see in a Hubble image. One part of the motivation for this design choice was the goal of creating aesthetically pleasing visualizations. Another motivation was based on collaborative work we were conducting with astrophysicists creating ensembles to study galaxy formation. Their real-world images inspired our interest in creating a galaxy-like visualization. A final motivation was to use our knowledge of the perceptual strengths and limitations of colour, texture, and motion in the human visual system to present perceptually optimal visualizations [6, 8].

The ensemble analysis we perform is independent of the visualization technique used to render the final results. This allows a user to replace our nonphotorealistic visualizations with more traditional approaches (*e.g.*, surfaces built with marching cubes, or ray-traced voxel visualizations) if these are preferred or considered more appropriate for the domain being analyzed.

Single Member Visualization

The star field visualizations can be viewed as a type of glyph-based volume rendering technique, used to display a single ensemble member m_i. Each glyph encodes member data from one octant in the octree. Let $D_i^r = \{d_1, d_2, ..., d_n\}$ be the set of data elements of m_i in octant o_r^l. The glyph g_r represents o_r^l as follows:

- The spatial location of g_r is the average location of the elements in D_i^r.
- The size of g_r represents n, the number of elements in D_i^r.
- The colour of g_r represents the average of the attribute values of the elements in D_i^r.



(a)

(b)

(c)

(d)

(e)

Figure 5: Multiple member visualization: (a) visualization of a four-member cluster: (b) first member, (c) second member, (d) third member, (e) fourth member

Our examples visualize temperature using a version of the rainbow colour scale that we perceptually corrected, with purples and blues for cold, greens for warm, and oranges and reds for hot [6]. This was driven by the physicists' expectation of this specific colourmap for representing temperature.

By default we render all leaf octants in the octree (Fig. 4a). To add flexibility, a viewer can define a threshold level $H_{\max}$ as the most detailed level in a visualization (Fig. 4b). This restricts the visualization to include octants from level $H_{\max}$ and all leaf octants above $H_{\max}$. By varying $H_{\max}$, scientists can take advantage of the hierarchical structure of the octree to visualize shape and data at different levels of detail. An octant from a more abstract level covers a larger 3D space, so its corresponding glyph provides a more abstract view, possibly removing distracting details that are not of interest. This also allows for member visualizations at different levels of detail without the need to rebuild the octree.

Multiple Member Visualization

Visualizing multiple members is one of the key differences between 3D ensemble visualization and traditional volume rendering. Multi-member visualization is necessary because ensemble analysis focuses not only on features of a single member, but also on shape and data relationships between members. One approach, used in [20], places members side-by-side with multiple linked views. This limits the number of members that can be compared, however, and assigns the responsibility for comparison to the viewer. Another solution is to overlay multiple members onscreen. This was shown in [17] to be inefficient and prone to visual clutter.

To address these issues, we designed a cluster visualization that extends the single member visualization to highlight similarities and differences in shape and attribute value distributions across multiple ensemble members. The visualization starts by performing an octant-by-octant summarization and comparison, creating a single glyph for each octant. It then uses the same strategies from the single member visualization to select a subset of octants to render (*i.e.*, to visualize at multiple levels of detail).

Assume $E' = \{m_1, m_2, ..., m_s\}, E' \subseteq E$, is a cluster of s members to visualize. Every $m_i \in E'$ with data in octant o_r^l is represented by an aggregated data point p_i^r in that octant. Let $P_i^r = \{p_1^r, p_2^r, \dots, p_n^r\}$ be the set of data points in o_r^l. The glyph g_r for o_r^l is created from the P_i^r as follows:

- The spatial location of g_r is the average location of the data points in P_i^r.
- The size of g_r represents n, the number of data points in P_i^r.
- The colour of g_r represents the average of the attribute values of the data points in P_i^r.
- Animation is introduced, with flicker frequency representing the variance of the attribute values of the data points in P_i^r.

Figure 5a shows a cluster visualization of four RHIC members presented individually in Figures 5b–e. Note the larger glyphs on either side of the dumbbell, and smaller glyphs at its center. This indicates that all four members have data on both sides of the dumbbell, but connect differently at the center. This can be seen in Figures 5b–e, where some members are strongly connected (Figure 5b), while others are only weakly connected (Figure 5c–e).

Hue provides additional insight into the average temperature of the members. It is high in the center of both ends of the dumbbell (orange and red glyphs), decreasing gradually toward the boundaries (blue glyphs). In the animated version of Figure 5a a small number of points in the center of the two dumbbell ends flicker more rapidly. This indicates higher temperature variances representing larger differences in temperature across the four members in these regions. This is a feature that is difficult to see in static, side-by-side visualizations (Figure 5b–e), but one that is clearly visible through the use of motion, a property that past research in our laboratory has shown to be perceptually effective at encoding information [8].

The cluster visualization highlights similarities by displaying overall shape and attribute distributions across members. It presents dissimilarities between members by visualizing member count and attribute variance at each octant with glyph size and flicker rate. Smaller glyphs indicate that fewer members have data in a given location. Glyphs flickering more rapidly represent larger attribute value differences in the region. Compared to techniques like [1], the cluster visualization is not meant to present detailed pairwise differences between members. Instead, it scales to visualize relationships between multiple members, allowing the scientists to trade off the number of members being visualized versus details for any single member. It is designed to cooperate with and validate the clustering results by highlighting regions of shape and data value similarity and difference within a cluster. For example, consider again the small glyphs in the center of Figure 5a that indicate the four members are connected differently. Scientists can choose a more detailed cluster in the cluster tree to separate the different shapes, or ignore the shape differences and continue to explore the four-member cluster.

RHIC Application

We collaborated with physics colleagues at Duke University to apply our methods to a RHIC ensemble with 224 members. This represents a more realistic ensemble size, compared to the smaller ensembles we used to demonstrate our analysis and visualization techniques. The physicists focused on differences in shape and temperature—an attribute value they are particularly interested in exploring.

Figure 6a visualizes an agglomerative cluster tree of the 224 RHIC members. It automatically identified two main clusters, encoded in the left and the right (Figure 6b) subtrees. Figure 6c is a cluster visualization of the 164 members from the left subtree. It includes members with connected dumbbell shapes, similar to the two example members shown in Figures 6e,g. Figure 6d is a cluster visualization of the 60 members from the smaller right subtree. It contains members with shapes that have two cones either separated or weakly connected, similar to the two example members shown in Figures 6f,h. The distribution of strongly versus weakly connected members was of particular interest to the physicists, since it highlighted the sensitivity of connectivity to small differences in input parameters. The physicists switched between the two cluster visualizations by selecting the left or right child of the root node in the cluster tree visualization. They also selected different leaf nodes in each subtree to visualize individual members and examine the quality of the clustering results.

A second exploration varied the number of clusters k from $k = 2$ (two clusters) to $k = 224$ (one member per cluster). Figure 7a plots the smallest dissimilarity for each k, that is, the *threshold* between the two most similar clusters.

The threshold jumps sharply at $k = 2$ to approximately 0.27. This is not surprising, since it represents the root node splitting into its left and right subtrees, but it does provide an indication of the amount of shape dissimilarity within the ensemble as a whole. The threshold falls to approximately 0.24 at $k = 3$, then slowly decreases until $k = 184$, when it falls to zero through $k = 224$. This indicates that the two most similar clusters contain identical shapes. The physicists investigated the inflection in the threshold at $k = 183$, and discovered that each cluster contained members with identical shapes. Figure 7 highlights the 183 clusters. Every subtree with a red root node represents a cluster, all of whose members are identical.

Physicist Feedback

Although we did not conduct formal experiments to compare task performance for the physicists' existing approaches versus our visualization system, we did discuss with them at some length their experiences from using our system.

Feedback was positive. Our colleagues noted that our system was much more efficient than their current approach of statistical and mathematical analysis with minimal visualization support. This was especially true when the physicists first looked at their results to obtain an initial understanding of an ensemble, and when they wanted to perform free-form exploration within an ensemble.

To compare the advantage of a basic visualization system

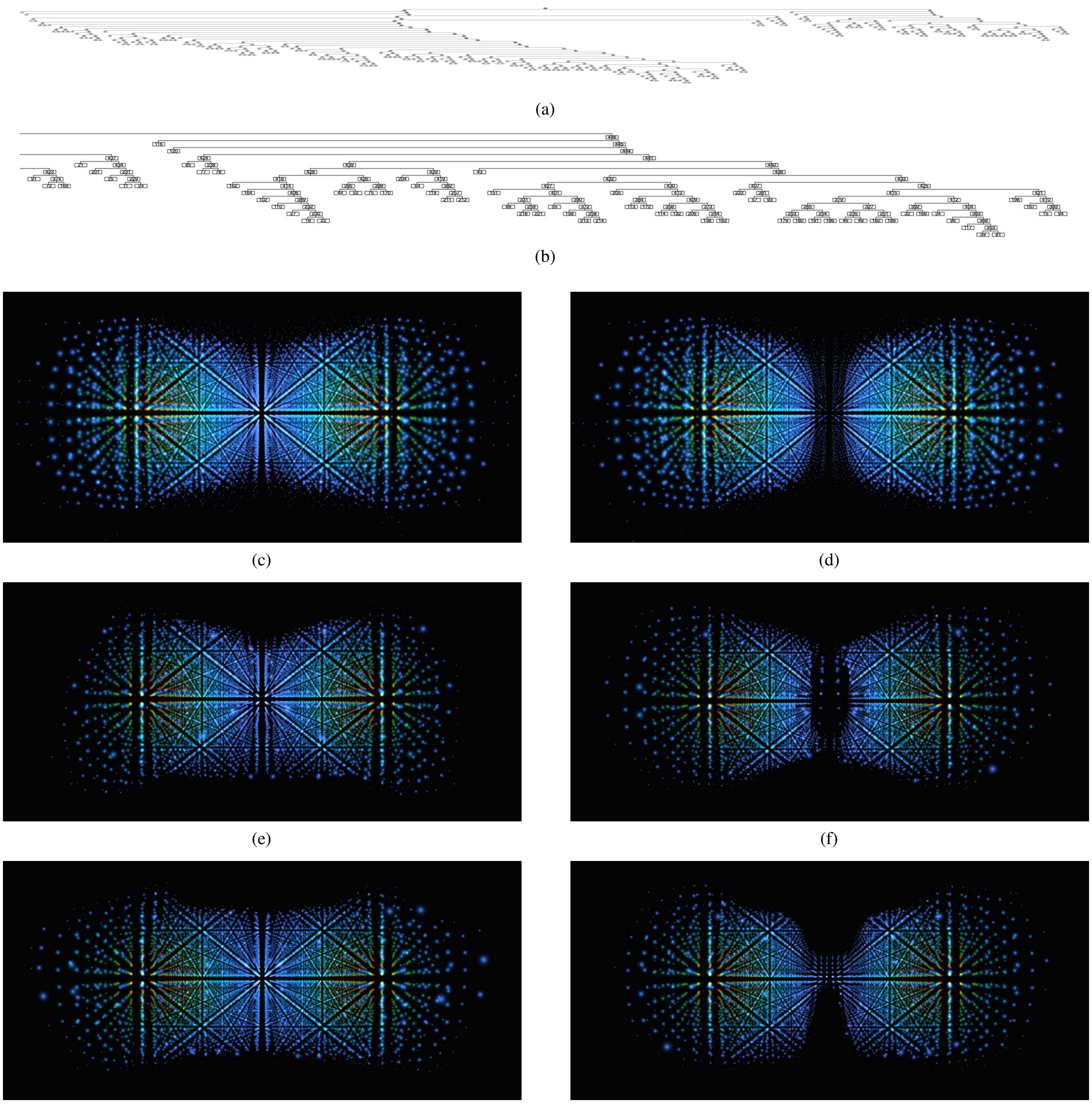
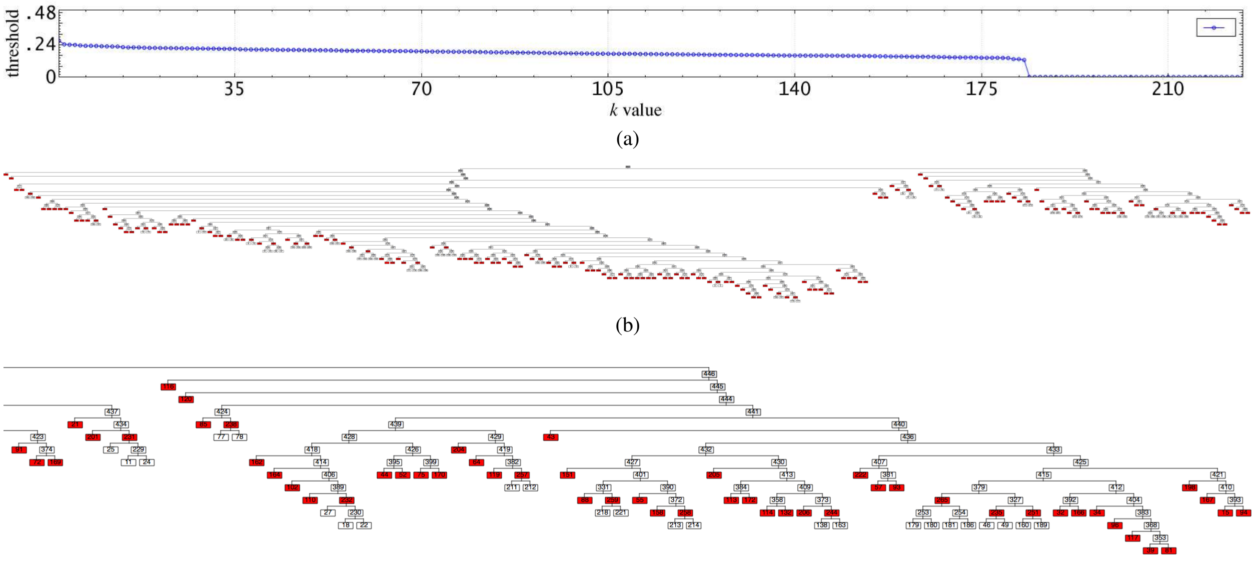


Figure 6: RHIC example: (a) agglomerative cluster tree of a 224-member RHIC ensemble; (b) close-up of the 60 members in the right subtree; (c) cluster visualization of the left subtree's 164 members, which are connected at the center; (d) cluster visualization of the right subtree's 60 members, which are disconnected at the center; (e,g) two single member visualizations from the left subtree; (f,h) two single member visualizations from the right subtree

alone versus our hierarchical technique, we asked the physicists to visualize members individually as a complete set, then try to identify members with similar shapes. Not surprisingly, the physicists found this difficult to do, especially when the ensemble contained numerous members. Moreover, subtle differences, for example, the variation in temperature internal to the dumbbell ends shown in Figure 5, were impossible to detect visually.

The physicists emphasized that the ability to cluster and visualize members offers important advantages to their current workflow, providing a way to rapidly explore within an ensemble to confirm expected findings and perhaps more importantly, to identify unexpected or unusual results. For example, determining that 75% of the ensemble members were strongly connected at the center and only 25% were weakly connected led to an investigation of how connectivity was related to changes in parameter inputs. This is important, since the physicists' over-



(a)

(b)

(c)

Figure 7: Varying k: (a) threshold for each k; (b) $k = 183$ clusters, each red cluster contains members with identical shapes; (c) close-up of the members in the right subtree

all goal is to choose parameters that mimic real-world results, and to understand whether slight variations in those parameters will lead to small or large changes in the simulation. The finding that forty different members have one or more identical partners, even when their parameter inputs are different, led to additional insights into member sensitivity to specific parameter and parameter range changes. The physicists noted that these findings would have been difficult and time consuming to identify using their existing data analytics algorithms.

Conclusions

We propose a framework to provide a scalable technique for analyzing ensembles. Our approach combines a level-of-detail hierarchical clustering algorithm to group similar members, a cluster tree visualization to provide an overview of inter-member relationships, and a detailed comparative visualization for single or multiple member clusters. The system allows scientists to start with a high-level overview, then zoom in to explore detailed shape comparisons between members.

We collaborated with physicists at Duke University to study RHIC ensembles. RHIC members can contain millions of data elements, producing CSV files up to 85 MB in size. Our octree representation reduces data size by aggregating data elements in an octant and linking spatially related elements. The system is capable of generating XML files that encode octree representations. For example, a 29 MB CSV file representing one member was converted to a 600 KB XML file encoding its octree representation.

Octree shape comparison mathematically captures dissimilarities between members, freeing a scientist from using visual perception alone to identify differences. Our enhanced algorithm guarantees accuracy even when large numbers of empty octants occur. It associates shape dissimilarities at multiple levels of abstraction in the octree, based on a scientist-chosen shape comparison ratio. The resulting cluster tree interactively guides scientists when they select members to visualize, increasing the efficiency of ensemble analysis.

Individual members and member clusters are visualized using a glyph-based approach. The technique is scalable. Including more members does not significantly increases the number of glyphs, so it will not lead to on-screen clutter or large increases in computation. Different ensemble views are integrated and coordinated, producing a multi-level, multi-perspective ensemble analysis system.

Our current visualizations focus more on general shape summarization versus detailed dissimilarity comparison. For example, they do not identify *which* members have data in a given octant. Multi-member visualizations should provide more powerful comparative details for in-depth dissimilarity analysis, perhaps by highlighting interesting sub-regions to avoid distraction and increase the efficiency of the analysis.

The system described here does not support the important need for temporal ensemble analysis, which if often required. We have recently investigated a number of more complex methods to identify temporal patterns within an ensemble (*e.g.*, cluster participation pattern mining and time-step pattern mining), with positive results [5].

The framework we propose is flexible and extensible. The system can be modified to analyze 2D spatial ensembles by replacing octrees with quadtrees. The dissimilarity calculation and cluster algorithms can be revised to meet domain requirements. The 3D visualization can be modified to adjust for specific features of interest. Given this, our future work focuses on improvements in each part of the framework, better coordination between the parts, and intelligent management of shape and data changes in the time dimension.

References

[1] O. S. Alabi, X. Wu, J. M. Harter, M. Phadke, L. Pinto, H. Petersen, S. Bass, M. Keifer, S. Zhong, C. G. Healey, and R. M. Taylor

II. Comparative visualization of ensembles using ensemble surface slicing. *Visualization and Data Analytics*, 8294(1):0U, 1–12, 2012.

[2] S. Busking, C. Botha, L. Ferrarini, J. Milles, and F. H.. Post. Image-based rendering of intersecting surfaces for dynamic comparative visualization. *The Visual Computer*, 27(5):347–363, 2011.

[3] E. Corchado and B. Baruque. WeVoS-ViSOM: An ensemble summarization algorithm for enhanced data visualization. *Neurocomputing*, 75(1):171–184, 2012.

[4] I. Demir, C. Dick, and R. Westermann. Multi-charts for comparative 3D ensemble visualization. *IEEE Transactions on Visualization and Computer Graphics*, 20:2713–2722, 2014.

[5] Lihua Hao, Christopher G. Healey, and Steffen A. Bass. Effective visualization of temporal ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):787–796, 2015.

[6] Christopher G. Healey and James T. Enns. Large datasets at a glance: Combining textures and colors in scientific visualization. *IEEE Transactions on Visualization and Computer Graphics*, 5(2):145–167, 1999.

[7] Christopher G. Healey and James T. Enns. Attention and visual memory in visualization and computer graphics. *IEEE Transactions on Visualization and Computer Graphics*, 18(7):1170–1188, 2012.

[8] Daniel E. Huber and Christopher G. Healey. Visualizing data with motion. In *Proceedings of the 16th IEEE Visualization Conference (Vis 2005)*, pages 527–534, Minneapolis, Minnesota, 2005.

[9] C. L. Jackins and S. L. Tanimoto. Oct-trees and their use in representing three-dimensional objects. *Computer Graphics and Image Processing*, 14(3):249–270, 1980.

[10] J. Kehrer and H. Hauser. Visualization and visual analysis of multifaceted scientific data: A survey. *Visualization and Computer Graphics, IEEE Transactions on*, 19(3):495–513, March 2013.

[11] P. Köthur, M. Sips, H. Dobslaw, and D. Dransch. Visual analytics for comparison of ocean model output with reference data: Detecting and analyzing geophysical processes using clustering ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 19:1893–1902, 2013.

[12] K. Matkovic, D. Gracanin, B. Klarin, and H. Hauser. Interactive visual analysis of complex scientific data as families of data surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1351–1358, Nov 2009.

[13] M. Mirzargar, R. T. Whitaker, and R. M. Kirby. Curve boxplot: Generalization of boxplot fo ensembles of curves. *IEEE Transactions on Visualization and Computer Graphics*, 20:2654–2663, 2014.

[14] M. Mukerjee. A little big bang. *Scientific American*, 280:60–65, 1999.

[15] B. Muller, J. Schukraft, and B. Wyslouch. First results from Pb+ Pb collisions at the LHC. *Annual Review of Nuclear and Particle Science*, 62:361–386, 2012.

[16] T. Nocke, M. Flechsig, and U. Böhm. Visual exploration and evaluation of climate-related simulation data. In *Proceedings of the 2007 Winter Simulation Conference*, pages 703–711, Dec 2007.

[17] M. N. Phadke, L. Pinto, O. Alabi, J. Harter, R. M. Taylor II, X. Wu, H. Petersen, S. A. Bass, and C. G. Healey. Exploring ensemble visualization. *Visualization and Data Analysis (VDA)*, 8294:0B, 1–12, 2012.

[18] H. Piringer, S. Pajer, W. Berger, and H. Teichmann. Comparative visual analysis of 2D function ensembles. *Computer Graphics Forum*, 31(3pt3):1195–1204, 2012.

[19] K. Potter, A. Wilson, P. Bremer, D. Williams, C. Doutriaux, V. Pascucci, and C. Johhson. Visualization of uncertainty and ensemble data: Exploration of climate modeling and weather forecast data with integrated ViSUS-CDAT systems. *Journal of Physics: Conference Series*, 180(1), 2009.

[20] K. Potter, A. Wilson, V. Pascucci, D. Williams, C. Doutriaux, P.-T. Bremer, and C. R. Johnson. Ensemble-Vis: A framework for the statistical visualization of ensemble data. In *IEEE International Conference on Data Mining Workshops (ICDMW '09)*, pages 233–240, 2009.

[21] Hanan Samet. *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann, San Francisco, California, 2005.

[22] J. Sanyal, S. Zhang, J. Dyer, A. Mercer, P. Amburn, and R. J. Moorhead. Noodles: A tool for visualization of numerical weather model ensemble uncertainty. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1421–1430, 2010.

[23] L. G. Tateosian, C. G. Healey, and J. T. Enns. Engaging viewers through nonphotorealistic visualizations. In *Proceedings 5th International Symposium on Non-Photorealistic Animation and Rendering (NPAR 2007)*, pages 93–102, San Diego, California, 2007.

[24] R. T. Whitaker, M. Mirzargar, and R. M. Kirby. Contour boxplots: A method for characterizing uncertainty in feature sets from simulation ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 19:2713–2722, 2013.

[25] A. T. Wilson and K. C. Potter. Toward visual analysis of ensemble data sets. In *Proceedings of the 2009 Workshop on Ultrascale Visualization (UltraVis '09)*, pages 48–53, 2009.

[26] J. Zhang and S. Smith. Shape similarity matching with octree representations. *Journal of Computing and Information Science in Engineering*, 9(3):034503:1–034503:5, 2009.

Author Biography

Born in China, Lihua Hao received her B.S. degree in Computer Science from Peking University, Beijing, China in 2010. In the same year, she joined the doctoral program in Computer Science at North Carolina State University (Raleigh, NC). Under the guidance of Dr. Healey, Lihua's research interests focus on data visualization and analytics. Her PhD thesis focused on scalable visual analytics of ensemble datasets, i.e., large correlated datasets collected from runs of a scientific simulation. While pursuing her degree, she also managed a research assistant funding project on web-based visualization to improve cyber situation awareness of network security data. Passed her PhD defense in Dec. 2014, Lihua is now working at Facebook Inc. (Menlo Park, CA) as a Research Scientist.