Effective Visualization of Temporal Ensembles

Lihua Hao, Christopher G. Healey, *Senior Member, IEEE*, and Steffen A. Bass

Abstract— An *ensemble* is a collection of related datasets, called *members*, built from a series of runs of a simulation or an experiment. Ensembles are large, temporal, multidimensional, and multivariate, making them difficult to analyze. Another important challenge is visualizing ensembles that vary both in space and time. Initial visualization techniques displayed ensembles with a small number of members, or presented an overview of an entire ensemble, but without potentially important details. Recently, researchers have suggested combining these two directions, allowing users to choose subsets of members to visualization. This manual selection process places the burden on the user to identify which members to explore. We first introduce a static ensemble visualization system that automatically helps users locate interesting subsets of members to visualize. We next extend the system to support analysis and visualization of temporal ensembles. We employ 3D shape comparison, cluster tree visualization, and glyph based visualization to represent different levels of detail within an ensemble. This strategy is used to provide two approaches for temporal ensemble analysis: (1) *segment based ensemble analysis*, to capture important shape transition time-steps, clusters groups of similar members, and identify common shape changes over time across multiple members; and (2) *time-step based ensemble analysis*, which assumes ensemble members are aligned in time by combining similar shapes at common time-steps. Both approaches enable users to interactively visualize and analyze a temporal ensemble from different perspectives at different levels of detail. We demonstrate our techniques on an ensemble studying matter transition from hadronic gas to quark-gluon plasma during gold-on-gold particle collisions.

Index Terms—Ensemble visualization

1 Introduction

An ensemble is a collection of data produced by a series of runs of a simulation or an experiment, each with slightly different initial conditions or parameterizations. Scientists from various disciplines use ensembles to simulate complex systems, explore unknowns in initial conditions, investigate parameter sensitivity, mitigate uncertainty, and compare structural characteristics of their models. Data collected from each run forms an *ensemble member*, or more specifically a *time-series ensemble member* if it contains results collected over a sequence of time-steps. Ensembles are difficult to analyze due to their large size and high complexity [27].

Different techniques have been developed for ensemble visualization. Some create a concise overview, but omit potentially important details from the original data [4, 21]. Others extend traditional visualization techniques for a single simulation to support comparison between members [1, 18]. This provides a better view of the individual members, but often limits comparison to a small member set. Seen in this way, the two main approaches to ensemble visualization provide either: (1) an overview that scales but at the expense of detail, or (2) a visualization that maintains detailed information but only displays a few members at a time.

We are collaborating with physicists at Duke University who are studying particle collision ensembles. Each simulated collision produces a time-series member of 3D particle data. The physicists are interested in studying how the shape of a volume evolves, where "shape" corresponds to the density and extent of the particles within the volume. This includes specific needs to: (1) explore within individual time-series members; and (2) compare both shape and data changes over time across multiple members. Currently, this is done by analyzing raw numeric data directly, or by producing simple visualizations of a single member at a single time-step. Our approach is designed to allow fluid exploration throughout the ensemble to identify and compare important shape and data features.

- *Lihua Hao is with NC State University. E-mail: lhao2@ncsu.edu.*
- *Christopher G. Healey is with NC State University. E-mail: healey@ncsu.edu.*
- *Steffen A. Bass is with Duke University. E-mail: bass@phy.duke.edu.*

Manuscript received 31 Mar. 2015; accepted 1 Aug. 2015; date of publication 20 Aug. 2015; date of current version 25 Oct. 2015. For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org. Digital Object Identifier no. 10.1109/TVCG.2015.2468093

More recent ensemble visualization systems support interactive analysis at different levels of detail [12, 19]. These systems rely on users to select a subset of members for detailed visualization, for example, by brushing in a high level ensemble overview. Currently, investigating ways to automatically capture relationships between members (inter-member relationships), or to explore important information related to changes in the time dimension, are open challenges.

We first propose a hierarchical approach to combine the two directions of ensemble analysis, focusing on static ensembles that do not change over time. Our technique reveals hierarchical inter-member relationships and supports visualization and analysis of one or more members simultaneously. An octree representation is built to compress the data and extract shapes from the ensemble [9, 22]. We extend the similarity matching in [28] to mathematically measure shape dissimilarity between member pairs. These dissimilarities are used to hierarchically cluster similar members into common groups. The result is visualized as a level-of-detail cluster tree that allows users to interactively perform comparative visualization among clusters of members with varying levels of shape and data similarity.

We next extend the static system to support analysis and visualization of temporal ensembles. This includes comparison of time-series members and pattern mining in the time dimension. Our extended system provides two approaches for temporal ensemble analysis: (1) *segment based;* and (2) *time-step based*.

The first technique, segment based analysis, combines similar shapes from all members across all time-steps. Time-series members are compressed into a sequence of member segments by combining neighboring member items with similar shapes. Dynamic time warping [24] is used to compare the compressed time-series members based on shape changes over time. Results are visualized as a cluster tree that highlights hierarchical inter-member relationships.

To explore common shape change patterns that occur at different time-steps and across different members, we cluster member segments over the entire ensemble, transforming it into a set of member cluster participation sequences. We adapt UpDown tree contiguous item sequential pattern mining (CISP) [3] to discover patterns in the participation sequences. The patterns identify contiguous shape changes and other important temporal features that occur frequently in the ensemble. We extend our multivariate visualization to display patterns and time-series member clusters, using animations to combine a sequence of visualizations ordered in time.

The second approach, time-step ensemble analysis, is motivated by our physics collaborators' need to compare members at each time-step. We independently clusters members at every time-step using our static

ensemble approach. Clustering the members converts the ensemble into a set of shape cluster participation sequences, one per time-step, suitable for closed CISP mining. This provides time-length reduction and optimal pattern matching that allows for relationship analysis at every time step.

2 Related Work

Various visualization techniques have been presented to support analysis of ensemble data, for example, with volume rendering, multidimensional visualization, and comparative visualization [2, 10, 16].

Ensemble-Vis was an early framework for visualizing weather forecasting and climate modeling ensembles [20]. It provides statistical aggregation and linked views for color maps, contours, height fields, and *spaghetti plots*, a technique for uncertainty visualization in meteorology that displays contours at specific attribute value boundaries. Noodles is a similar visualization tool built for analysis of meteorological ensembles [23]. It provides more complex statistical aggregation and uncertainty measurements, employing circular glyphs, ribbons, and spaghetti plots for data visualization.

Follow-on research extended ensemble visualization to support efficient visual comparison between pairs of members. Ensemble Surface Slicing (ESS) [1] color-codes member surfaces and slices them into equal-width strips, then builds a combined representation by extracting and visualizing strips member-by-member. Visual discontinuities between strips highlight surface shape differences between the strips' members. Phadke proposed pairwise sequential animation for 3D ensembles by constructing glyphs for data elements whose color and shape represent attribute value and parent member. Glyph opacity varies over time based on a series of transfer functions to compare pairs of members [18].

Recent studies have proposed systems that support multi-level and interactive visual analysis. Matkovic developed a system to investigate multi-run simulation results as families of data surfaces, plotted in 2D [12]. The system uses multiple linked views to analyze data surface projections and aggregations at a top level with parallel coordinates and scatterplots; at a second level with parallel coordinates and function graphs; and at a low level with 2D height 3D surface views to explore a selected surface. Piringer designed an interactive system to compare 2D function ensembles [19]. The system includes a domain-specific overview to combine features into a 2D heatmap, a member-specific overview to visualize members in a scatterplot, and a detailed member view to visualize small subsets of members in a 3D scatterplot.

Whitiker and Mirzargar proposed contour and curve boxplots to visualize statistical properties, outliers, and variability in ensembles of contours or curves [13, 26]. Band depth statistically summarizes the centrality of members of an ensemble, which are visualized using specialized boxplots. Demir developed multi-charts, an overlay of bar and line charts to present statistical properties of ensemble members [5]. Köthur studied temporal properties of ensembles, generating temporal profile clusters for different members [11], then consolidating them to identify profiles representing specific features of interest.

Different approaches to characterize shape also exist. Gyulassy applied the Morse-Smale complex to segment a volume into ordered topological features of interest based on gradient flow [6]. Pascucci described the use of contour trees to represent topological changes in level sets [17]. He proposed a visualization technique based on an orrery to allow interactive filtering of topological properties (minima, maxima, saddle points, etc.) at multiple levels of detail.

Ensemble visualization research builds on previous techniques like glyph based rendering, comparative analysis, charts, and linked views. Current systems support sophisticated statistical methods to represent uncertainty [13, 26], to summarize member data in ways that highlight areas of interest [5], and to consider the temporal aspects of ensemble members [11].

We propose an octree based ensemble visualization framework that measures shape similarities between members, then combines correlated members into a level-of-detail cluster hierarchy for detailed member comparison. The hierarchy is visualized as a cluster tree, providing a visual representations for multiple members. Since our physics collaborators must compare changes in 3D shapes—spatial distributions of particles—over time and identify frequent spatial and temporal patterns in their particle collision ensembles, we further extend our static technique to analyze level-of-detail relationships within and between time-series members using CISP mining, time-series clustering, and animation based visualization of changes in shape and data over time in a member cluster or temporal pattern.

Compared to existing ensemble analysis techniques, our approaches enable users to analyze and visualize temporal ensembles from different perspectives, and at user-chosen levels of details. This provides the following novel contributions:

1. Shape and data comparison for 3D ensemble members.
2. Cluster tree visualizations to highlight hierarchical member relationships, and to support user-chosen tradeoffs in the number of members visualized versus individual member detail.
3. Pattern identification for both static and temporal ensembles.
4. Individual and multi-member visualizations for shape, data, and pattern comparison.

3 Static Ensemble Visualization

We define an ensemble $E = \{M_1, M_2, \ldots M_N\}$ with N time-series members $M_i \in E$. M_i is a sequence of *member items* $(m_{i,1}, m_{i,2}, \ldots m_{i,T})$ with T time-steps. Static ensemble visualization analyzes members at a specific time-step $E_t = \{m_{1,t}, m_{2,t}, \ldots m_{N,t}\}$, providing an overview of shape similarity between members, presented as a cluster tree. Clusters in the tree are interactively selected and visualization for detailed exploration of a set of members' shapes and data values.

Our techniques assume a spatial distribution of data samples. More abstract data can still be processed if a spatial layout can be imposed prior to analysis. This is similar to many information visualization techniques, which start with a layout algorithm to position data elements, followed by the application of a more traditional visualization approach.

Octree construction. We begin by constructing an octree O [9, 22] to compress member data and extract shapes at different levels of detail. O's root node is the minimum bounding cube that covers all data elements in E. For each m_i, we recursively subdivide parent octants into eight equal-sized child octants until the number of elements within an octant meets an upper bound $O_{\max}$. Following construction, each octant contains data from q members. Data in the octant is aggregated to encode: (1) q data points representing the average spatial location of each m_i's data elements; and (2) q pairs (μ_i, σ_i) representing the average and variance of the attribute values in m_i's data elements.

Shape dissimilarity. We use m_i's octree to define the *shape* of a member as the spatial distribution of its data elements. This lays a foundation for hierarchical overviews of inter-member shape relationships, freeing users from predicting relationships between members apriori to choose which subsets of members to analyze and visualize.

Our shape comparison algorithm extends octree shape similarity matching for 3D shape retrieval [28]. To support shape clustering, we measure dissimilarity between members, as opposed to similarity. We modify [28] to maintain dissimilarity accuracy for octrees with large common empty regions. Given m_i and $m_j \in E$, let $dis^r_{i,j}$ be the dissimilarity 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 are the number of data elements of m_i and m_j that lie within o^l_r. We consider m_i and m_j to be equivalent if $cnt^r_i = cnt^r_j$ ($dis^r_{i,j} = 0$), as completely different if o^l_r is empty for either m_i or m_j ($dis^r_{i,j} = 1$), and as partially different otherwise, measured as:

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

The range of $dis^r_{i,j}$ is $[0,1]$, with higher scores for a larger relative differences in point counts. Next, we aggregate $dis^r_{i,j}$ between m_i and

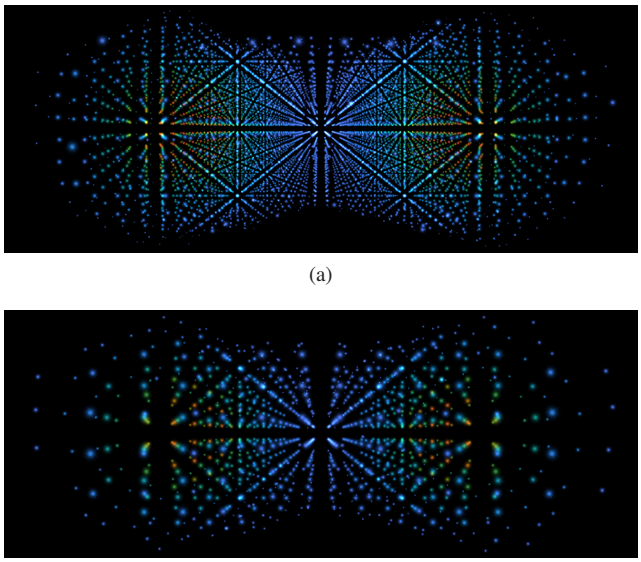


Fig. 1. Member visualization: (a) a member visualized in full detail; (b) an aggregated visualization at the fourth level of the member's octree

m_j at each octree level l with N^l non-empty octants (Eq. 2 left) to produce a level dissimilarity $dis^l_{i,j}$. Finally, we aggregate $dis^l_{i,j}$ over all levels, starting at the root, to create an overall dissimilarity score $dis_{i,j}$ between m_i and m_j based on H, the height of O (Eq. 2 right).

$$dis^l_{i,j} = \frac{\sum_{r=1}^{N^l} dis^r_{i,j}}{N^l} \qquad dis_{i,j} = \frac{\sum_{l=1}^{H} w^l dis^l}{\sum_{l=1}^{H} w^l} \tag{2}$$

A weighting factor $w^l = 1/\gamma^l$ weights the dissimilarities at different levels in O according to a shape comparison factor γ, assigning larger weights to more detailed levels.

Visualization. $dis_{i,j} \forall i, j$ produces a dissimilarity matrix. This is used as input to an agglomerative clustering algorithm that builds a cluster tree whose visualization reveals inter-member relationships. Users interact with the cluster tree to choose individual members or clusters—subsets of members—to examine in detail. Member data is visualized using glyphs that highlight shape and data differences (Fig. 1). Size represents the number of data elements in each octant, color represents a user-selected data attribute value, and flicker visualizes the attribute value's variance over the octant's members (*e.g.*, Fig. 3's visualization of individual members, and the combined visualization they produce). As members' average attribute values differ over a wider range, the octant's glyph flickers between fully opaque and transparent more quickly (see Supplemental Video 1).

Our choice of the specific colors, sizes, and flicker rates were selected based on the perceptual properties of these features. Guidelines from perceptual experiments conducted in our laboratory have identified the strengths and limitations of each of the features, both in isolation and when combined in a common visualization [7, 8].

Our approach extends traditional multivariate visualization to support general shape visualization and region-by-region comparative visualization across multiple static ensemble members. This provides a detailed view of shape, data element distributions, and important attribute value differences. Users interactively choose the level of similarity when exploring inter-member relationships, deciding when to explore shape and data overviews, and when to visualize fine-grained details in individual member datasets. The contributions of our static ensemble approach are:

1. A scalable technique to represent an ensemble as a level-of-detail cluster tree based on shape similarity.
2. A glyph visualization that supports visual comparison of both shape and data across multiple members.

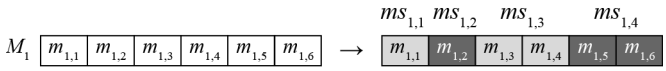


Fig. 2. Converting M_1's six members $(m_{1,1}, m_{1,2}, m_{1,3}, m_{1,4}, m_{1,5}, m_{1,6})$ into four member segments $(ms_{1,1}, ms_{1,2}, ms_{1,3}, ms_{1,4})$

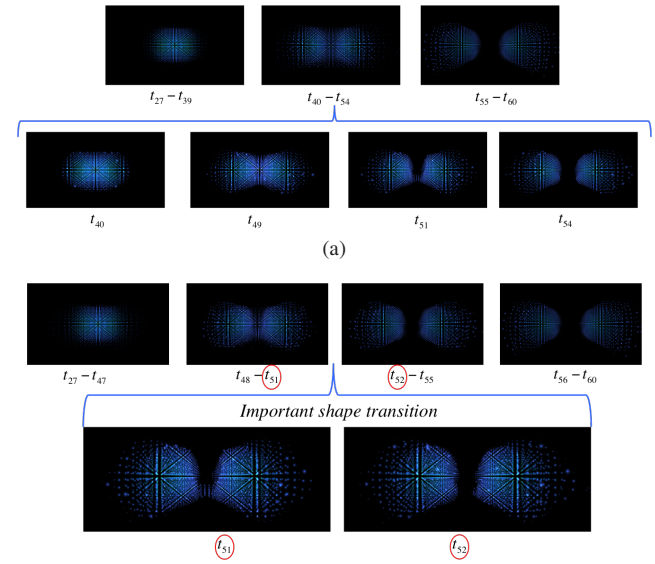


Fig. 3. Member segmentation: (a) ratio based, the second member segment contains member items with significantly different shapes; (b) shape integration based, time-steps t_{51} and t_{52} where the volume splits are correctly identified as a boundary between two member segments

3. An interface that allows users to interactively vary the level of similarity of the members under examination.

4 Temporal Ensemble Visualization

We next extend our static techniques to include analysis and visualization of a temporal ensemble E. Our approach supports comparison, exploration, and visualization of changes in shape and data over time.

4.1 Member Segmentation

Including a time dimension leads to a significant increase $N \rightarrow N \times T$ in the total number of member items, requiring $N^2 \times T^2$ pairwise shape comparisons for a brute-force shape clustering. To improve efficiency, we combine similar member items into *member segments* (Fig. 2) and move our dissimilarity evaluation from the member item level to the member segment level. *Member segmentation* groups similar shapes adjacent in a local region in time within M_i, abstracting its T member items to T' member segments $(ms_{i,1}, ms_{i,2}, \ldots ms_{i,T'})$, $T' \leq T$. This can significantly improve performance if $T' \ll T$ by reducing the number of octree comparisons to $O(N^2 \times T'^2)$.

4.1.1 Shape Integration Based Segmentation

We initially implemented a contribution ratio based member segmentation inspired by color image segmentation. In a temporal ensemble, each member consists of temporally ordered results, so shapes collected at adjacent time-steps are often similar and therefore may belong to the same member segment. We represent a member as a one-dimensional row of member items (pixels), with neighbors defined as the items that immediately precede and follow a target item.

Member segmentation uses the first item in the segment as a reference item, recursively updating the median distance using dissimilarities between the reference item and new items. Unfortunately, there is no guarantee that the first item is a good representative of a segment. Fig. 3a converts a member that starts with a cylinder shape and ends with two separated cones over 34 time-steps into 3 member segments,

using $r = 1$ and a dissimilarity threshold of 0.4. Because of the way reference items were chosen, the second member segment incorrectly combines the time-steps that contain the separation event.

The key to improving the quality of member segmentation is to find a measure of median distance that considers the dissimilarity between all member items, but without significantly increasing the cost of octree comparisons. The octree shape integration dissimilarity measure in Section 4.2 compares two groups of member items with one octree comparison. We apply this shape integration technique as follows:

1. Select a member item $m_{i,t}$ that is not part of any member segment, with neighbors $m_{i,t-1}$ and $m_{i,t+1}$.
2. If a neighbor does not belong to any existing segment and is within a user defined dissimilarity threshold of the median of items in the segment, add it to the segment.
3. Update the segment's median dissimilarity using the dissimilarity between an integrated octree of the current segment's items and the new item.

$$dis_{\text{med}} = dis_{\text{shape-integration}}(ms_{\text{curr}}, m_{\text{new}}) \tag{3}$$

4. Recursively consider neighbors of new items until no more items can be added to the segment. At the end of the recursion, a member segment for $m_{i,t}$ is generated.
5. Repeat steps 1 through 4 until every member item is assigned to a member segment.

Shape integration segmentation uses the average shape of a segment as its median to determine whether a new item should be assigned. It does not requires a contribution ratio, so the quality of the segmentation depends only on median distance. Fig. 3b uses shape integration to abstract the same 34 member items in Fig. 3a using the same dissimilarity threshold of 0.4. This produces four member segments, properly capturing the volume's split between t_{51} and t_{52}. Each member segment represents a smooth shape change with limited shape variations from the median at each time-step, producing higher quality segments that capture major shape changes in the time dimension.

Our member segmentation approach efficiently combines neighboring member items. It requires $O(T)$ octree comparisons to segment a member with T time-steps. Shape clustering requires $O(T^2)$ octree comparisons to generate the dissimilarity matrix and $O(T^2)$ time for hierarchical clustering. Assuming T' is the average number of member segments per member, segmentation groups the $N \times T$ member items in E into $N \times T'$ member segments. This simplifies analysis and visualization of the ensemble by reducing the length of each member in the time dimension. T' will depend on characteristics of the data and the user-chosen dissimilarity threshold.

4.2 Cluster Shape Dissimilarity

To efficiently compare sets of member items, we propose an *octree shape integration* approach that measures the dissimilarity between the member items' integrated octree representations. This extracts the overall shape of member items in a set by averaging their data in each octant. Analogous to Eq. 1 for static ensemble analysis, let $cnt^l_{p,r}$ and $cnt^l_{q,r}$ be the number of data points of $m_{i,p} \in Mi$ at time-step p and $m_{j,q} \in M_j$ at time-step q, for octant o^l_r at level l in the octree. We measure overall shape dissimilarity between M_i and M_j at o^l_r as follows:

$$dis^l_r = \frac{\left| cnt^l_{M_i,r} - cnt^l_{M_j,r} \right|}{\max(cnt^l_{M_i,r}, cnt^l_{M_j,r})}$$
$$cnt^l_{M_i,r} = \frac{1}{|M_i|} \sum_{p \in M_i} cnt^l_{p,r} \qquad cnt^l_{M_j,r} = \frac{1}{|M_j|} \sum_{q \in M_j} cnt^l_{q,r} \tag{4}$$

By measuring dissimilarity between M_i and M_j at each level of the octree and aggregating the results across user-specified starting and stopping levels, we create a final dissimilarity score, similar to the shape dissimilarity calculation for a static ensemble.

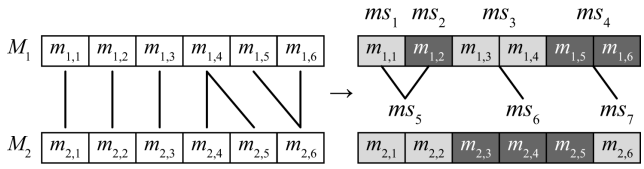


Fig. 4. Improving dynamic time warping using member segments

If the shapes of member items in each cluster are similar, octree shape integration provides a more precise approximation of cluster shape dissimilarity, since it respects the overall shape of each cluster by averaging their octree representations. If the shape dissimilarity between each member item pair is unknown and a large number of member items are involved in the comparison, the shape integration technique is still efficient because it traverses the octree representation a constant number of times: twice for shape averaging and once for the final dissimilarity measure. The complexity of the algorithm does not increase significantly as the size of the clusters increase.

5 Segment Based Ensemble Visualization

Segment based ensemble visualization is a more general approach that captures hierarchical relationships across all member items, proposes member segmentation to reduce length in the time dimension, and captures possible shifting in the time dimension.

5.1 Time-Series Member Clustering

Time-series member clustering addresses one important goal in ensemble visualization: identifying similar simulation or experiment runs. In a temporal ensemble, analysis of inter-member relationships (*i.e.*, sets of similar members) is extended from a static level between $m_{i,t}$ and $m_{j,t}$ to a sequential level between M_i and M_j. Since changes in shape can happen at different time-steps and over different periods of time, we use *dynamic time warping* (DTW) [24] to find an optimal alignment between two members. We then construct a cluster tree that visualizes member similarity in E at different levels of detail.

DTW generates a non-linear alignment between two temporal sequences that minimizes pairwise distances by shifting and distorting in the time dimension. Dynamic programming is used to identify an optimal minimum distance between M_i and M_j by building the shortest warping path in a $T \times T$ matrix D where $D(p,q)$ encodes the shape dissimilarity between $m_{i,p}$ and $m_{j,q}$. It requires $O(T^2)$ shape comparisons to calculate the dissimilarity between M_i and M_j, and $O(T^2 \times N^2)$ shape comparisons to compare all members in E.

We use the same T' member segments built in Section 4.1 to improve performance by measuring dissimilarity between M_i and M_j and finding the optimal match between the two corresponding sequences of member segments that minimizes their differences (Fig. 4). It costs $(T^2 \times T'^2)$ time to compare all pairs of segment sequences, a significant improvement when the octree is large and $T' \ll N$.

DTW generates a matrix encoding all pairwise member dissimilarities in a time-series ensemble, respecting contiguous changes in shape over time. We use this dissimilarity matrix as input to agglomerative clustering, producing a cluster tree that supports inter-member relationship analysis and visualization.

5.2 Member Cluster Participation Pattern Mining

A second important requirement during ensemble visualization is to identify shape patterns: common changes in shape over time. To identify shape changes that occur across multiple members, we need to mark similar shapes. We do this by generating an $(N \times T') \times (N \times T')$ dissimilarity matrix of member segments. This matrix is used to build a segment cluster tree that allows users to choose clusters that combine similar shapes across members.

Shape clusters and member segments both group member items based on changes in shape over time. The key difference is that member segments occur within a single member. Shape clusters can occur across multiple members (Fig. 5).

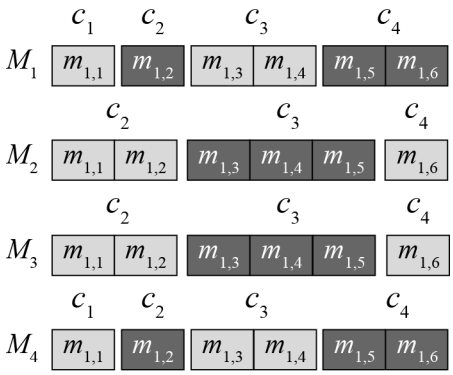


Fig. 5. Member cluster participation sequences mark common changes in shape c_1, c_2, c_3, and c_4 that occur across multiple members

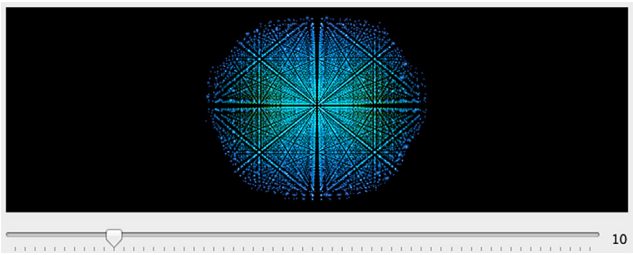


Fig. 6. Visualizing a time-series member that fades member items in and out based on time-step, the slider denotes the current t

As an example, consider an ensemble containing 14 member segments $(ms_1, ms_2, \ldots ms_{14})$ that are grouped into 4 shape clusters (c_1, c_2, c_3, c_4) based on shape change similarity over time (Fig. 5). Now, ensemble members can be transformed into sequences of shape clusters, for example, $M_1 = (m_{1,1}, m_{1,2}, m_{1,3}, m_{1,4}, m_{1,5}, m_{1,6}) \Rightarrow S_1 = (c_1, c_2, c_3, c_4)$. We call this transformation *segment cluster abstraction*, and the resulting sequence S_1 a *member cluster participation sequence*. Transforming $E = \{M_1, M_2, \ldots M_N\}$ produces N member cluster participation sequences $\{S_1, S_2, \ldots S_N\}$.

The number and length of member cluster participation sequences are often large, making it infeasible to manually identify frequent patterns across the S_i. To solve this, we use an UpDown tree to perform CISP mining [3]. This allows us to automatically discover common subsequences in the S_i. These patterns represent a shape change that occurs across multiple members. Once patterns are identified, they can be visualized using our glyph based visualizer. It is also possible to visualize *where* in each member a common pattern occurs. This is discussed in detail when we introduce the RHIC ensemble in Section 7.

5.3 Temporal Visualization

To visualize shape change, we extend our glyph based visualization to use animation to display changes in shape over time. A sequence of member items fade in and out based on their temporal order, generating a movie of contiguous changes in shape. A slider below the visualization captures the time-step of the current visible member item, cluster, or pattern (Fig. 6). To improve the efficiency for visualizing long members, we support an abbreviated representation that displays a sequence of member segments rather than the original member items (see Supplemental Video 2).

Visualizing member clusters from Section 5.1 produces a sequence of visualizations of multiple members, where the p-th visualization reveals shape similarities and dissimilarities between the members at the p-th time-step. Analogously, a pattern visualization is a sequence of visualizations of multiple members containing the given pattern.

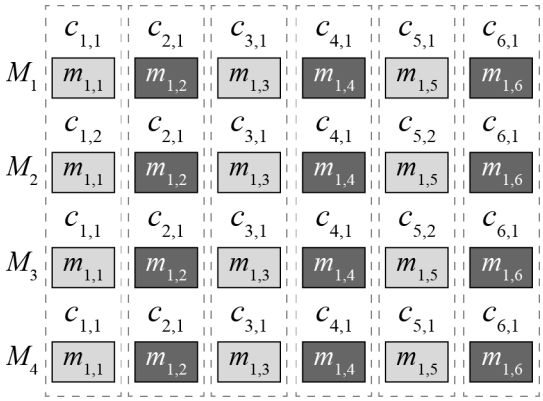


Fig. 7. Time-step cluster participation sequences mark similar shapes within a given time-step, *e.g.*, $c_{5,1}$ and $c_{5,2}$ in time-step 5

6 Time-Step Based Ensemble Visualization

Another approach, advocated by our physics collaborators, compares members at each time-step. To meet this goal, we designed time-step based ensemble analysis, similar to the segment based approach but embedding a static ensemble analysis at every time-step.

Our time-step ensemble analysis assumes that time-series members are exactly aligned in the time dimension. It compares member items at each time-step using Manhattan distance. The pairwise member similarities serve as input to construct a member cluster tree that provides an overview of inter-member relationships at the given time-step.

To discover important patterns that occur across multiple members, users define a threshold to select shape clusters across all time-steps. This converts the time-series members into a set of member cluster participation sequences, allowing us to apply CISP to identify common shape changes that occur frequently across multiple members. We use our temporal visualization from Section 5.3 to visualize the member clusters and the patterns they form.

6.1 Time-Series Member Clustering

As in segment based analysis, the goal of time-series member clustering is to identify similarities between simulation or experiment runs. Given members that are aligned in time (Fig. 7), we adapt Manhattan distance to compare members by averaging the dissimilarities at all time-steps.

$$dis(M_i, M_j) = \frac{\sum_{t=1}^{T} dis(m_{i,t}, m_{j,t})}{T} \quad (5)$$

Dissimilarity matrices at every time-step have been generated during time-step member clustering (Section 5.1), so it costs $O(T)$ time to compare two time-series members. The comparison is more efficient than DTW dissimilarity calculations, but only applicable if members are aligned in time. Comparing all pairs of time-series members generates an $N \times N$ dissimilarity matrix. We apply agglomerative clustering to build a cluster tree that identifies inter-member relationships throughout the ensemble.

6.2 Time-Step Cluster Participation Pattern Mining

Time-step pattern mining addresses the goal of identifying common changes in shape over time across members. Octree construction and shape dissimilarity calculations in time-step based ensemble analysis are identical to segment based analysis. To combine similar shapes among members and capture common subsequences, we cluster members within each time-step. This produces T cluster trees, each capturing the hierarchical inter-member relationships at a given time-step. Because we restrict analysis to time-steps independently, we do not combine similar shapes that are located in different time-steps, and we do not reduce the lengths of the time-series members. It requires $O(N^2 \times T)$ octree comparisons to analyze an $N \times T$ ensemble. This suggests that time-step shape clustering is most applicable for ensembles with short time-series members aligned in the time dimension.

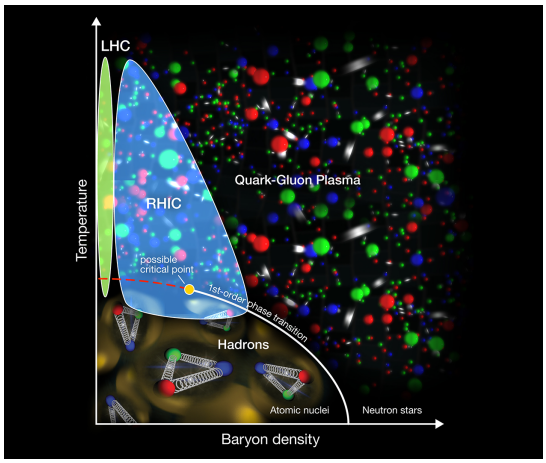


Fig. 8. Transition from nuclei to free quarks and gluons; protons and neutrons are disintegrated at extremely high temperature or density [25]

Similar to the segment cluster abstraction in Section 5.2, we transform a time-series member into a time-step shape cluster participation sequence to perform closed pattern mining (Fig. 7). A time-step shape cluster $c_{t,j}$ is identified by a time-step t and a cluster ID $j \in t$. We again use an UpDown tree CISP analysis to identify frequent changes in shape over time. A time-step cluster consists of similar shapes at the same time-step, so a pattern occurs in every M_i starting at the same time-step t_p and ending at the same time-step t_q.

6.3 Temporal Visualization

Member cluster and shape change pattern visualization are identical to the animation based temporal visualization in Section 5.3. Time-step shape clusters do not combine similar shapes at different time-steps, so we do not support abstracted member visualization. The visualization reveals more detailed shape changes over time but can be less efficient, especially for long time-series.

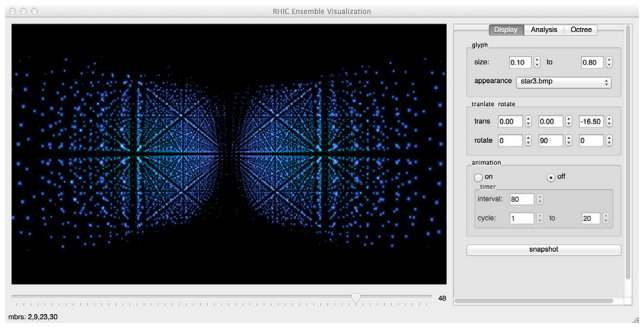
7 Practical Application

We implemented the ensemble analysis techniques as a stand-alone system. We then applied both segment based and time-step based temporal ensemble analysis to a simulated RHIC ensemble with 41 members $\{M_1, M_2, \ldots M_{41}\}$, each with $T = 60$ time-steps. Members contain from 180,000 to 3.3 million particle samples. The analysis focuses on differences in shape and temperature—an attribute that our physics collaborators are particularly interested in.

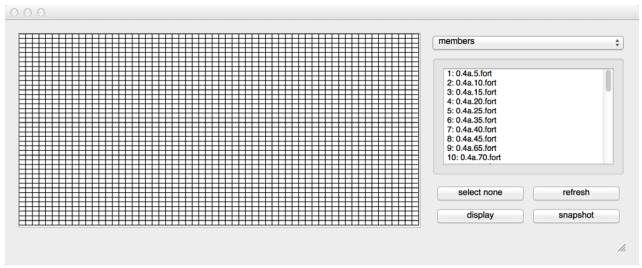
7.1 RHIC Ensemble

We are collaborating with physicists at Duke University to study quark–gluon plasma (QGP) formation using the the Relativistic Heavy Ion Collider (RHIC) at Brookhaven National Laboratory. Heavy ion collisions at very high energies are used to investigate interacting matter under extreme temperatures far above those of normal nuclear matter [15, 25]. Quantum chromo-dynamics (QCD), the quantum field theory of strong interactions, confirms matter transition from a hadronic gas to a quark–gluon plasma at extremely high temperatures and energy densities. In a QGP phase, protons and neutrons separate, releasing quarks and gluons (Fig. 8).

Theoretical physicists believe that quark–gluon plasma existed in the universe during the first few microseconds after its creation in the Big Bang. Our collaborators use the RHIC to collide two opposing beams of gold nuclei while they are traveling at relativistic speed [14]. The resulting collisions generate extremely hot, dense bursts of matter and energy that recreate QGP conditions similar to the very early universe. Because of the extremely short time and length scales of the heavy-ion collision, scientists must use indirect means to extract conclusions from RHIC data, in our case, by comparing the measurements to computational models of the collision dynamics. As with all simulations, a key challenge is the choice of initial conditions for the collision model, which is based on relativistic fluid dynamics.



(a)



(b)

Fig. 9. Stand-alone visualization system for temporal 3D ensembles: (a) volume visualization of one or more members on the left, user interface on the right; (b) ensemble overview grid representing all time-steps

Ensembles were collected using a RHIC simulation that models hydrodynamic calculations of a gold on gold collision at a center of mass of 200 GeV per nucleon. The ensembles contain hundreds of members from simulation runs with slightly different quantum fluctuations of protons and neutrons, start times for the hydrodynamic calculations, and granularities of the initial energy–density deposit that enters the hydro field. Members contain large numbers of particles positioned in 3D, each encoding separate data attributes like temperature, energy density, pressure, and velocity.

The physicists stated specific needs to: (1) explore within an individual time-series member; and perhaps more importantly (2) compare shape and shape evolution across multiple members, to understand how parameter choices and parameter sensitivity affect simulation results. We discuss below how our segment based and time-step based analysis approaches are used to address these research questions.

7.2 Ensemble Analysis System

We constructed a stand-alone visual analytics system consisting of a 3D volume visualization widget and user interface (Fig. 9a), and an ensemble overview widget (Fig. 9b) to highlight members and time regions participating in selected segmentation, clustering, and pattern mining results. Elements in the volume represent aggregated particle sample positions and attribute values for the given octant across all members being visualized. The aliasing artifacts in some of the volumes are a consequence of the regular grid sampling used in the underlying simulation. The overview widget visualizes E as an $N \times T$ grid, each row representing a member and each column representing a time-step (member item). When users select one or more segments, clusters, patterns or members in the checklist, the corresponding member items are highlighted. Selecting in the grid generates a tooltip with the member ID, time-step ID and file name of the corresponding member item. Double clicking in the grid visualizes the corresponding member item in the visualization widget.

The system can input data points from CSV files, or octree representations from a XML files, where each file corresponds to a time-step in a member. Based on users' requests, the system can perform any of the temporal ensemble analysis approaches from Sections 5 and 6, or it can load and visualize pre-calculated segmentation results and dissimilarity matrices.

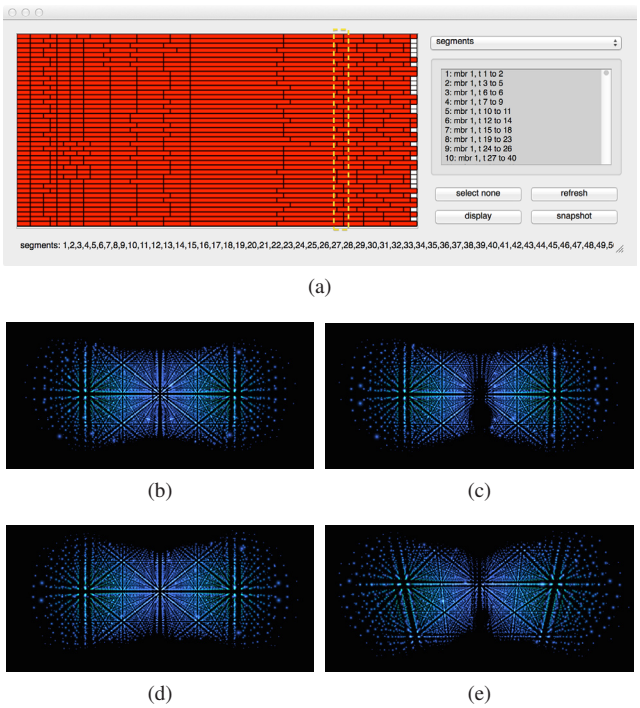


Fig. 10. Shape integration segmentation: (a) all member segments selected and visualized in red, the dashed yellow rectangle identifies cluster boundaries where cylinder-shaped members start to split into dumbbell shapes: (b,c) split at t_{48}–t_{49} in M_3; (d,e) split at t_{49}–t_{50} in M_5

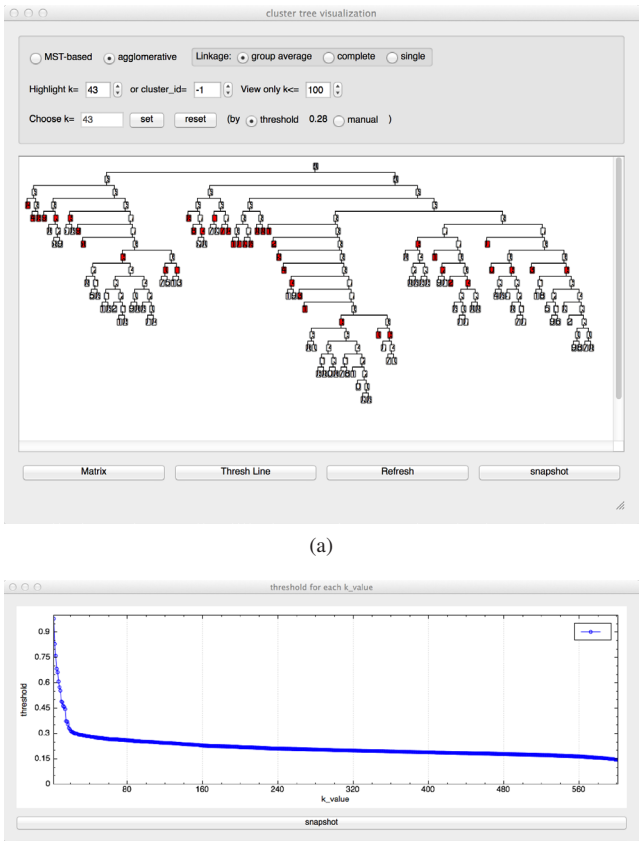


Fig. 11. (a) Visualizing the agglomerative segment cluster tree with $k = 100$ cluster nodes; (b) dissimilarity between the two most similar clusters for different values of k

7.3 Segment Based Ensemble Analysis

We applied shape integration member segmentation to the RHIC ensemble, using a dissimilarity threshold of 0.35. The 41×60 member items were assigned to 601 member segments $(ms_1, ms_2, \ldots ms_{601})$. Fig. 10a visualizes the segmentation results in the ensemble overview when all the segments in the list are selected. Each red rectangle represents a segment. A short segment indicates a rapid change in shape over time, while a longer segment indicates smooth and slow changes in shape. A RHIC simulation terminates when specific constraints are satisfied, so not all members have the same number of time-steps. White grid cells in the last column of Fig. 10a indicate that the corresponding RHIC members have terminated. A good segmentation result captures important shape transition time-steps in the simulations. For instance, segment boundaries at the time-steps circled by the yellow dashed rectangle identify the time-steps when cylinder shapes start to transition into dumbbells, visualized in Fig. 10b-e.

The system mathematically captures shape dissimilarities for all segment pairs according to shape integration based octree comparisons (Section 4.2). The resulting dissimilarity matrix produces a member segment cluster tree with 1,201 nodes. The system allows users to choose how much of the tree to visualize by defining a number of cluster nodes k or a cut-off similarity threshold σ. For example, Fig. 11a visualizes the last $k = 100$ nodes generated during clustering. Fig. 11b graphs the thresholds of the cluster results by k, that is, the dissimilarity of two most similar clusters. This helps users choose an appropriate k or σ value. Based on Fig. 11b and feedback from our physics colleagues, we set $\sigma = 0.28$ to assign segments into $k = 43$ clusters, highlighted by the red nodes in Fig. 11a.

Based on the 43 clusters, the system transforms each time-series member to a member cluster participation sequence using segment cluster abstraction (Section 5.2), applying UpDown tree CISP to explore shape changes in the ensemble. The ensemble overview contains a list of member cluster participation patterns, sorted by their occurrence frequencies and lengths. Selecting patterns in the list highlights the member items covered by these patterns. Fig. 12 shows the top five patterns. The three red regions identify the times when shape changes throughout the ensemble are similar. Note that patterns three and four (cs_{1045} and cs_{1059}) are contained in pattern five. The white grid cells inside the middle red area indicate that member M_{24} is different from all the other members in this time region.

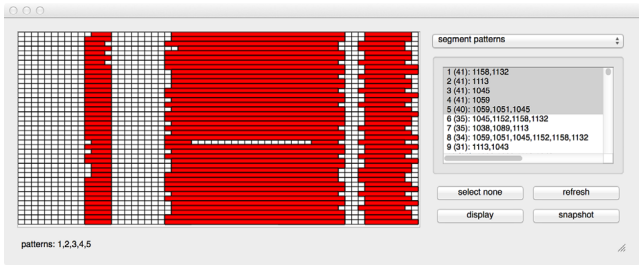
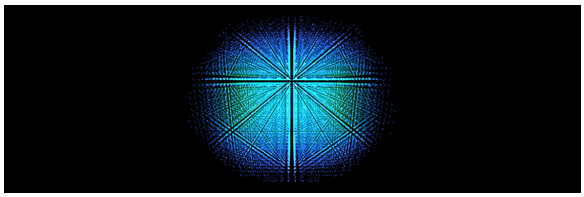
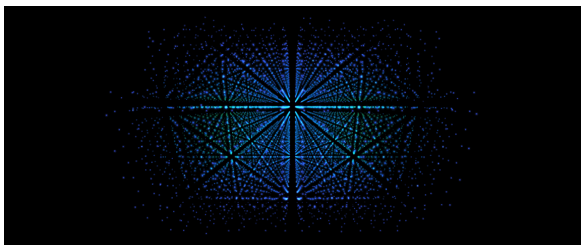


Fig. 12. The five longest and/or most frequent segment patterns, highlighted in red in the overview grid

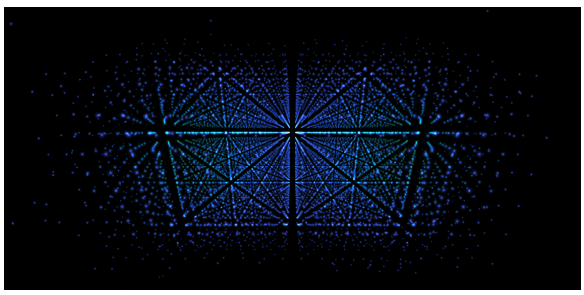
The system supports visualization of one pattern at a time. For example, visualizing pattern five fades in and out visualizations of segment clusters cs_{1059}, cs_{1051} and cs_{1045}, respectively (Fig. 13). The pattern covers approximately 25 time-steps, but the visualization summarizes the data as three segment cluster visualizations. Segment pattern visualizations capture a general understanding of major changes in shape over time, omitting more detailed shape differences. Pattern mining relies on the segment clustering abstraction results from Section 5.2. Too fine a clustering assigns similar segments to different clusters, ignoring common shape changes that would otherwise be captured. Too coarse a clustering combines dissimilar segments and results in patterns that encode dissimilar shape changes. Using different segment clustering results to perform pattern mining enables users to analyze shape changes at different levels of abstraction.



(a)



(b)



(c)

Fig. 13. Member cluster participation pattern five contains three cluster segments: cs_{1059}, cs_{1051}, and cs_{1045}

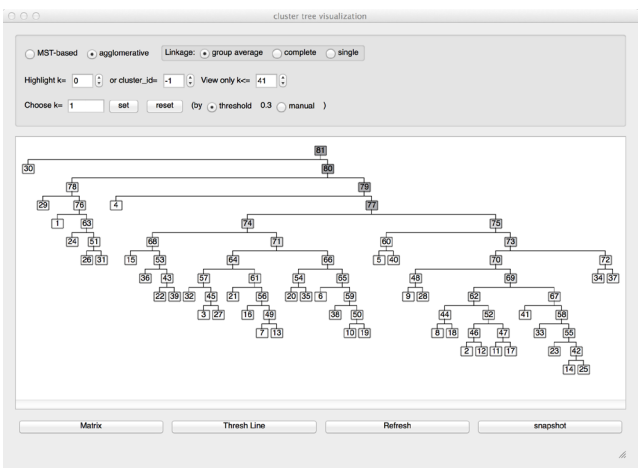
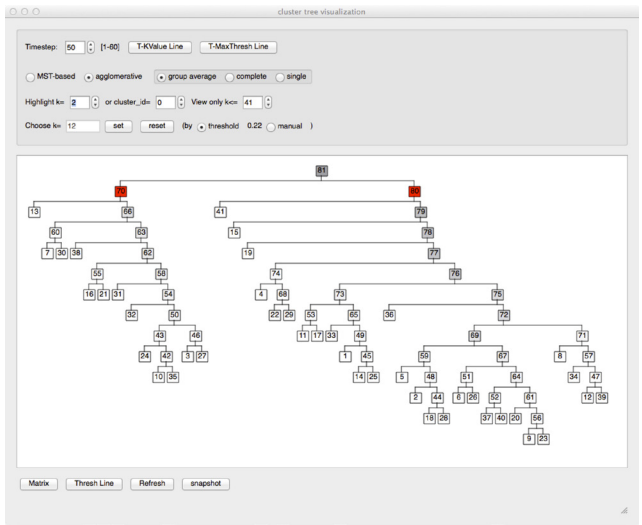
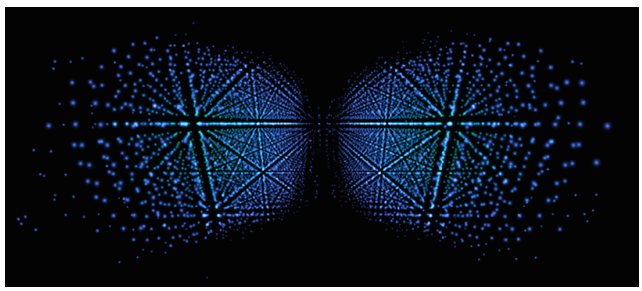


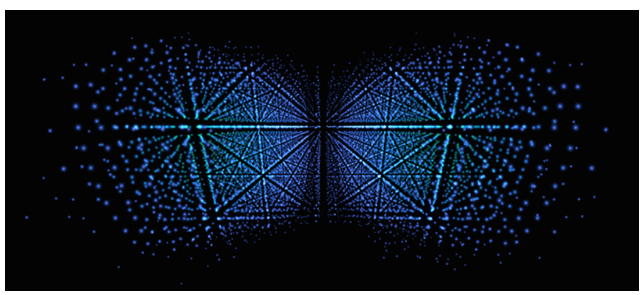
Fig. 14. DTW time-series member cluster tree



(a)



(b)



(c)

Fig. 15. (a) Cluster tree visualization at t_{50} with $k = 2$ clusters highlighted: (b) disconnected cones cluster; (c) connected dumbbell cluster

Users may want to compare the similarity of entire members, rather than member segment or shape cluster-level details. To do this, our system computes dissimilarities between every pair of members using DTW. Fig. 14 visualizes a DTW member cluster tree, where nodes contain similar members. This can be used together with, or in place of, a finer member segmentation approach, which leads to more detailed member comparisons, but at the expense of higher time complexity during segmentation and segment clustering.

7.4 Time-Step Based Ensemble Analysis

We next applied time-step ensemble analysis to the 41-member RHIC ensemble. We first calculated dissimilarities between all member pairs at every time-step to produce 60 cluster trees encoding static hierarchical inter-member relationships. Fig. 15a visualizes the cluster tree at t_{50}. Double-clicking the two red nodes in the tree visualizes the two member clusters (Fig. 15b,c). Fig. 15b combines members in t_{50} that are shaped like two separated cones. Fig. 15c combines members in t_{50} with a connected dumbbell shape. The static cluster trees enable scientists to analyze inter-member relationships at different levels of detail at a particular time-step.

The system compares time-series members with Manhattan distance (Section 6.1). Fig. 16 visualizes the time-series member cluster tree, an overview of inter-member relationships with members aligned over time.

To identify common shapes across members, we cluster members at every time-step. Fig. 17a visualizes the maximum and average thresholds of the clustering results encoded in the 60 cluster trees. This allows scientists to choose a threshold to specify the number of clusters at each time-step. For this application, we choose a cut-off threshold of $\sigma = 0.22$. Fig. 17b visualizes the number of clusters at every time-step. From t_1 to t_8 and t_{49} to t_{52}, members are assigned to a large number of

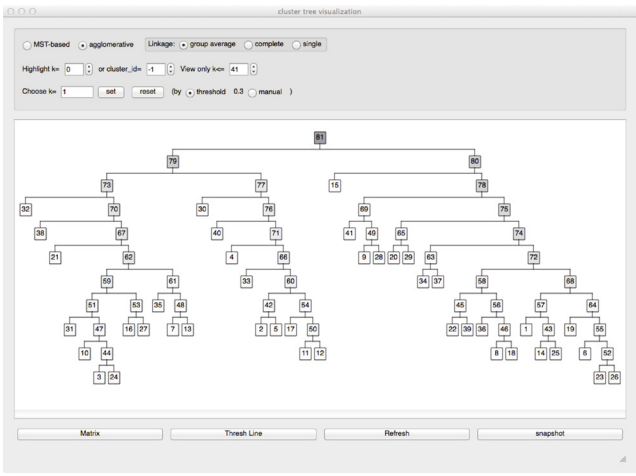
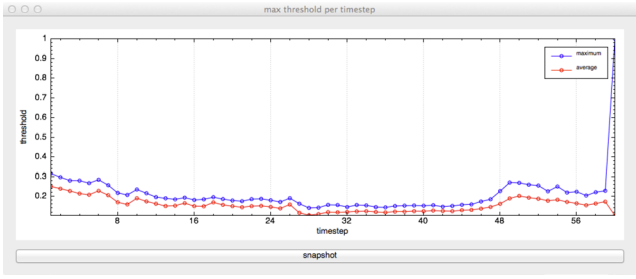
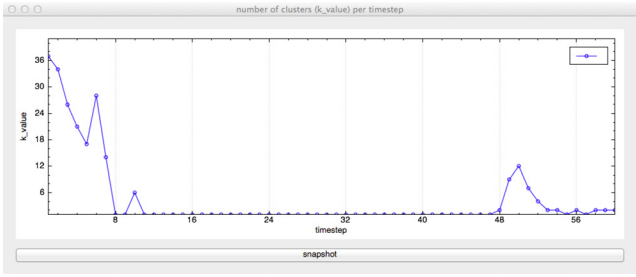


Fig. 16. Time-step comparison based member cluster tree



(a)



(b)

(c)

Fig. 17. Member cluster results at each time-step: (a) maximum and average clustering thresholds; (b) number of clusters; (c) shape change patterns common to all members

clusters, indicating larger shape dissimilarities in these time-steps.

Given the clustering results, the system transforms all members into a set of time-step shape cluster participation sequences, using time-step shape abstraction (Section 6.2). It then performs CISP to explore shape changes that occur frequently in the ensemble. Fig. 17c highlights patterns that occur in all 41 members. Users can apply different cluster results to discover patterns at different levels of abstraction. Shape changes covered by a pattern always start and end at the same time-steps, since the members are aligned in time. The length of a time-step pattern is equivalent to the number of time-steps covered by the pattern, providing a more detailed summarization and comparison of common shape changes.

8 Conclusions

The techniques described in this paper provide a scalable, level-of-detail analysis framework for temporal ensembles. We present a high-level overview of inter-member relationships, allowing users to choose the amount of detail to visualize during ensemble exploration and member comparison. Analysis of the ensemble is not limited to a particular time-step, since it identifies changes in shape across time. Users can analyze the ensemble from different perspectives, comparing results from both segment based and time-step based analysis.

Member segmentation locates both smooth and rapid shape variations in ensemble members, capturing the time-step boundaries containing these important changes. This allows users to visualize abstract shape changes summarized as sequences of member segments. We apply octree integration to efficiently compare segments. A cluster of segments represents similar shapes in different members and at different time-steps. The resulting cluster tree enables analysis and visualization at user-chosen levels of detail and time.

Pattern mining highlights similar shape changes across multiple members. Member cluster participation sequences identify patterns that address shifting and distortion in the time dimension. Time-step shape cluster participation sequences capture common shape patterns at a more detailed level, based on members directly aligned at each time-step. Pattern visualization summarizes important shape changes as clusters of member segments or time-step by time-step pattern occurrences. The member item grid visualizations highlight time regions and members covered by selected segments, clusters, or patterns, enabling users to rapidly focus on time periods or members of interest. In this way, different ensemble views are integrated and coordinated, providing a multi-level analysis system.

Feedback from our physics colleagues has been uniformly positive. The system design was influenced by their needs (*e.g.* the inclusion of time-step based analysis), and they have initiated integration of our tools into their workflow to study additional ensembles they have constructed.

In summary, we present a set of techniques that support a scalable approach to analyzing temporal ensembles. Our methods combine clustering and pattern mining based overviews with detailed shape and data comparison using animated glyph visualizations. This offers the following contributions:

1. Shape and data comparison for ensemble members (Sec. 3).
2. Cluster tree construction to highlight hierarchical member relationships, and to support user-chosen tradeoffs in the number of members visualized versus individual member detail (Sec. 3).
3. Common pattern identification for both static and temporal ensembles (Sec. 4, 5, 6).
4. Glyph based visualization for shape data, and pattern comparison (Sec. 3, 5.3, 6.3).

Performance improvements are critical in large data analytics. The time complexity of our temporal ensemble analysis is dominated by: (1) I/O time; and (2) the number of octree shape comparisons performed. For example, the 41 member RHIC ensemble octree contains more than 20,000 octants, making shape comparison computationally expensive. Our current system enables users to reload previous segmentation results and dissimilarity matrices, but we must still regenerate the matrices if new shape comparison or segmentation metrics are applied. To improve performance, we intend to explore approaches to reduce the number of shape comparisons, and to use parallel processing to expedite dissimilarity calculations.

The quality of the segmentation results during shape integration depend on shape comparison, starting time-step, and the measure of median dissimilarity. Users may need to investigate different values to make an appropriate choice. To improve effectiveness, we intend to explore automatic evaluation methods to choose optimal parameters.

The current system does not automatically choose the number of clusters to present from the cluster tree. Users must select a k value to transform members into sequences. Member visualization and cluster visualization are designed to integrate with the cluster tree visualization, meaning they are meant to visualize the overall shape of a cluster of members, in addition to revealing detailed shape similarities. We will explore techniques to overcome these limitations.

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 (VDA 2012)*, 8294(1):0U, 1–12, 2012.

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

[3] J. Chen. Contiguous item sequential pattern mining using UpDown tree. *Intelligent Data Analysis*, 12(1):25–49, 2008.

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

[5] 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.

[6] A. Gyulassy, V. Natarajan, V. Pascucci, P.-T. Bremer, and B. Hamann. Topology-based simplification for feature extraction from 3d scalar fields. In *Proceedings of the 16th IEEE Visualization Conference (Vis 2005)*, pages 535–542, Minneapolis, Minnesota, 2005.

[7] C. G. Healey and J. 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.

[8] D. E. Huber and C. 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. *IEEE Transactions on Visualization and Computer Graphics*, 19(3):495–513, 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, 2009.

[13] M. Mirzargar, R. T. Whitaker, and R. M. Kirby. Curve boxplot: Generalization of boxplot for 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. *arXiv preprint 1202.3233*, 2012.

[16] T. Nocke, M. Flechsig, and U. Bohm. Visual exploration and evaluation of climate-related simulation data. In *2007 Winter Simulation Conference (WSC 2007)*, pages 703–711, Washington, D.C., 2007.

[17] V. Pascucci, K. Cole-McLaughlin, and G. Scorzelli. The Toporrery: Computation and presentation of multi-resolution topology. In T. Möller, B. Hamann, and R. D. Russell, editors, *Mathematical Foundations of Scientific Visualization*, pages 19–40. Springer, New York, NY, 2009.

[18] 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 2012)*, 8294:0B, 1–12, 2012.

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

[20] 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):012089, 2009.

[21] 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, Miami, FL, 2009.

[22] H. Samet. *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann, San Francisco, CA, 2005.

[23] 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.

[24] P. Senin. Dynamic time warping algorithm review. Technical Report CSDL-08-04, University of Hawaii, Honolulu, HI, 2008.

[25] K. M. Walsh. Tracking the transition of early-universe quark soup to matter-as-we-know-it. `http://www0.bnl.gov/rhic/news2/news.asp?a=4473&t=today`, 2014.

[26] 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.

[27] 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, Portland, OR, 2009.

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