Comparative Visualization of Vector Field Ensembles Based on Longest Common Subsequence

Richen Liu [1*] Hanqi Guo [3†] Jiang Zhang [1‡] Xiaoru Yuan [1,2§]

1) Key Laboratory of Machine Perception (Ministry of Education), and School of EECS, Peking University
2) Beijing Engineering Technology Research Center of Virtual Simulation and Visualization, Peking University
3) Mathematics and Computer Science Division, Argonne National Laboratory

Abstract

We propose a longest common subsequence (LCSS)-based approach to compute the distance among vector field ensembles. By measuring how many common blocks the ensemble pathlines pass through, the LCSS distance defines the similarity among vector field ensembles by counting the number of shared domain data blocks. Compared with traditional methods (e.g., pointwise Euclidean distance or dynamic time warping distance), the proposed approach is robust to outliers, missing data, and the sampling rate of the pathline timesteps. Taking advantage of smaller and reusable intermediate output, visualization based on the proposed LCSS approach reveals temporal trends in the data at low storage cost and avoids tracing pathlines repeatedly. We evaluate our method on both synthetic data and simulation data, demonstrating the robustness of the proposed approach.

1 Introduction

A simulation ensemble is a set of simulations generated from models with different initial values and boundary conditions. Ensemble pathlines are a set of pathlines traced in the vector fields of different simulation members. Domain scientists are interested in the regions of vector field ensembles with high variation or high similarity across different ensemble runs. The accuracy of the similarity values depends on the distance between ensemble pathlines. If defined properly, the distance metric can provide meaningful and expressive comparative visualization results and help scientists discover and highlight the characteristics of the simulation models under different parameter conditions.

Existing similarity measurements to reveal variation presented in ensemble vector fields fall into two types. The first is pointwise Euclidean distance [7, 22] (we abbreviate it to pointwise distance in this paper). The other is linearized deformation measure [10] on vector field ensembles. The pointwise method defines the distance between ensemble pathlines by accumulating their pointwise distances along timesteps. It requires time series data of equal length. In general cases, truncating the longer series or padding zeros to the shorter one is performed to meet the equal-length constraint . More important, the pointwise distance is often sensitive to small perturbations or outliers. However, simulation data are always noise-prone because of the inaccuracy of data generation and collection. Thus, outliers and missing data are not rare. A small perturbation often induces severe misalignments between point pairs and exaggerates the measured distances. The longer the pathline is, the larger the magnification that will be introduced. The misalignment

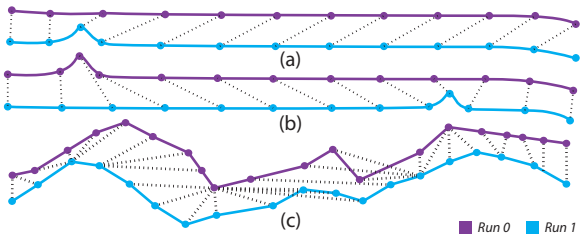

Figure 1: Pointwise misalignments caused by small perturbation or asymmetric variation in velocity direction: a single point in just one pathline (a); shifting occurrence time of a wavelet (outlier) (b); a 2D illustration of dynamic time warping.

can be caused by just one point with a small perturbation in the velocity direction or by just one shifting occurrence time of a wavelet (outlier), as shown in Figure 1a and Figure 1b. Furthermore, the exaggerated and unpredictable distances substantially enlarge the distribution range of the distance between ensemble pathlines, resulting in data in higher dynamic range.

In addition to the pointwise method, a new metric [10] was proposed to measure linearized deformation within a given Lagrangian neighborhood in ensemble vector fields. It employs the principal component analysis (PCA) to measure shape deformations for computational fluid dynamics simulations. One drawback of the PCA-based method is that the large joint variance cannot accurately and fully describe the variation in ensemble vector fields, because it reveals the linearized shape deformation derived from the flowmap. Moreover, the PCA-based method estimates the variance over a finite amount of time. It needs to trace pathlines repeatedly to compute joint variance among different time ranges. Ensemble analysis is thus expensive because of the computationally intensive pathline tracing, if scientists want to explore the ensemble behavior among different time ranges. Furthermore, pathlines can take 1,000 times more space than the unsteady vector field itself [7], which usually poses formidable challenges to handle and visualize the ensemble vector field data because of the multiple simulation members.

Arguably, scientists have different needs when studying ensemble behavior. They may be more concerned about whether the ensemble runs go through the common regions and may ignore the small timestamp differences of the peer elements on ensemble pathlines. For example, Mirzargar et al. [16] have designed three ensemble pathline parameterizations to visualize the variability. One of their parameterizations, the arc-length parameterization, focuses on comparing the geometry locations and ignores small timestamp differences. Therefore, the traditional pointwise comparison along the pathline timestep is an overconstrained condition. It is available for using more elastic distances in order to get more expressive comparative visualization.

Dynamic time warping (DTW) and pointwise Euclidean distanc are two widely used distance measures for time series data [21]. DTW is an elastic distance measure, which solves the problem of local time shifting of time series and can work with time series

*e-mail: richen@pku.edu.cn
†e-mail: hguo@anl.gov
‡e-mail: jiang.zhang@pku.edu.cn
§e-mail: xiaoru.yuan@pku.edu.cn (corresponding author)

of different lengths. However, all the elements in DTW must be matched, even the outliers, as illustrated in Figure 1c. The beginning and end timesteps in one series are always mapped to the beginning and end timesteps of the other series [13], which is overconstrained and inflexible. A small portion of outliers in the beginning and the end of the timesteps often leads to incorrect results [21]. Moreover, computing the DTW distance is expensive, because the time-consuming L_p norm must be computed.

To remedy these problems in traditional pointwise and DTW distances, we design an adaptive distance quantification method based on the longest common subsequence (LCSS) [9], to measure the similarity by quantifying the common passing regions shared by different ensemble runs. The proposed LCSS-based method has four benefits. First, it can work with time series of different lengths. Second, it is robust to small perturbations and outliers, compared with pointwise distance. Third, it is less sensitive to sampling timesteps, because it does not force to match the end points in the DTW distance. Fourth, it is more efficient than DTW and thus more suitable for measuring the distance of ensemble pathlines. It allows multiscale temporal comparison without repeatedly tracing pathlines, and it takes lower storage cost for temporal trend analysis.

2 Background

In this section, we first define a pathline in a 3D vector field. Then we review the few existing literature on the visualization of vector field ensembles and line comparison. We consider a pathline P in 3D ensemble vector fields, which consists of a set of consecutive points, $P = \{\langle x_1, y_1, z_1, t_1\rangle, \langle x_2, y_2, z_2, t_2\rangle, \cdots, \langle x_s, y_s, z_s, t_s\rangle\}$, where s is the number of points. Because the evolution time of the ensemble pathlines is nonuniform, we need to resample along the timestep. After that, the last dimension of the timestep can be reduced as follows, where t is the number of the resampled points.

$$P' = \{\langle x_1, y_1, z_1\rangle, \langle x_2, y_2, z_2\rangle, \cdots, \langle x_t, y_t, z_t\rangle\} \quad (1)$$

2.1 Visualizations of Vector Field Ensembles

Ensemble pathlines are produced by sets of ensemble runs simulated by varied models or under different parameter constraints on the ensemble vector fields. The analysis of ensemble pathline is of great significance to simulation science applications such as climate change, operational weather forecast, and computational fluid dynamic.

One of the classic methods for comparing ensemble vector fields is the pointwise method. This method is used in many recent works [7, 22]. It measures distance or summarizes feature between ensemble pathlines by their discretized points, which are sampled along the evolving timestep of the pathline or preprocessed by some specific parameterization. Guo et al. [7] use the pointwise distance to compare ensemble pathlines. They release the seeds evenly at each fixed location in ensemble fields, where a set of pathlines are integrated until they reach the end of their lifetime or go out of the field domain. The 2D line bands are summarized by pointwise locations in the contour boxplot method [22], which was designed to reveal the uncertainty in a 2D simulation data. Inspired by contour boxplot, a more generalized tool named curve boxplot [16] has been proposed to extend to 3D or even higher dimensions in order to extract the variability. Theoretically, they both leverage the mathematical notion of data depth, which can help reveal how central a line instance is within the distribution of the ensemble members. Based on the contour boxplot and curve boxplot, a novel technique named streamline variability plots [5] has been proposed to show the clustering trends of the ensemble streamlines.

To quantify the linearized deformation presented in Lagrangian neighborhood for each fixed locations in ensemble fields, Hummel et al. [10] proposed a PCA-based method for computational fluid dynamics simulations. This method can evaluate both individual and joint transport variance to reveal the characteristics of the ensembles. Most high values of joint variance are expected in regions with strongly varying transport behavior across the runs of the ensemble. The shape changes of the neighborhood cannot fully represent the variation among ensemble pathlines because they place more emphasis on the sensitivity formed within the Lagrangian neighborhood. Jarema et al. [11] have designed interactive similarity matrices and glyphs to compare the vector field ensembles; however, this comparison is limited to a 2D scenario. Another classic shape deformation evaluation approach for vector field is FTLE [8]. Uncertainty visualization in ensemble data can illustrate the characteristic features of the simulations. Uncertainty presented in vector field ensembles can be quantified by standard deviation, interquartile range, and width of 95% confidence interval [18]. Shen et al [19] proposed a framework to do data-level comparison. It can reveal subtle differences between two datasets with different grid resolutions through intermediate mesh. If observation data is available, it can estimate the predictive uncertainty [6], which can help identify potential outlier runs. Clustering is one of most powerful techniques used to extract the uncertainty or other ensemble behavior [20, 3]. The histogram clustering method is used in an interactive tool named Multi-Charts [3]. It allows to explore the ensemble data at multiple levels (focus and context).

2.2 Line Comparison

Measuring the line similarity is a fundamental problem in visualization, including the fieldline similarity in vector fields and the curve similarity in general data. Fieldline similarity in vector fields can be used to extract their statistical features, especially for their uncertainties and the summarized geometric information. Previous approaches to modeling the similarity between time series include the use of the Euclidean and DTW distance [21]. A similarity quantification approach [2] based on pointwise Euclidean distances for measuring streamline proximity has been used to select streamlines near interesting flow features, such as critical points and separations. Vlachos et al. [21] have proposed an LCSS-based method to discover similar tracking features of animals or humans. However, this method places more emphasis on the shape similarity between two time series data instead of the block-based geolocation differences in ensemble simulations. Moreover, tracking data is different from ensemble data: the former is closely related to human behavior. The measurement of curve similarity is an indispensable step in curve comparison. Chambers and Wang [1] developed a novel curve similarity measure between homotopic curves based on how hard it is to deform one curve into another one continuously; they defined the minimum possible surface area swept by a homotopy between the curves. Recently, Liu et al. [15] designed a sketch-based method to extract user-defined derived feature (i.e., vortex line) from vector field ensembles and compare them in some linked views.

3 Our Method

In this paper, we employ a novel metric, namely longest common subsequence, to measure the distance between ensemble pathlines. The LCSS distance measures how many common blocks the ensemble pathlines pass through and computes their similarity through the number of shared blocks. Figure 2 illustrates the pipeline of our proposed approach. Initially, the pathlines are traced by numerical integration, namely, the Runge-Kutta method. Our algorithm is parallelized with a MapReduce-like framework [12] in order to efficiently handle large-size ensemble data. Then the parallel LCSS sequence encoding and the parallel LCSS distance computation are integrated in the framework. The LCSS sequence encoding scheme is to encode the pathlines into LCSS sequences, which are the input parameters of the LCSS algorithm. The block indices (i.e., LCSS sequence codes) are saved to support multiscale

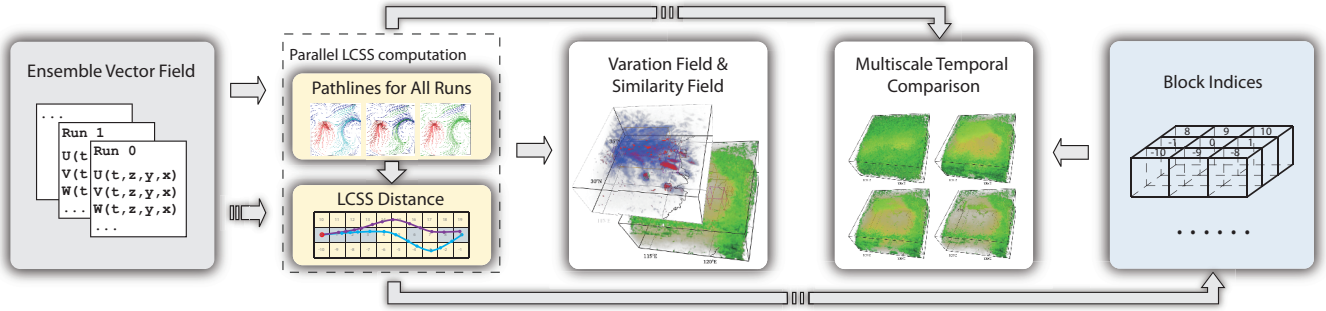

Figure 2: Pipeline of our work. We firstly trace pathlines in parallel from the raw ensemble data. By employing the parallel LCSS sequence encoding and distance metric, the generated pathlines then are encoded into LCSS sequences for further visualization and multiscale temporal comparison.

temporal comparison at lower storage cost and without repeatedly tracing the pathlines. For the final visualization part, we use a GPU-based volume rendering method to show the variation field and similarity field, following the method in [14].

3.1 LCSS-Based Distance Measure

The LCSS algorithm finds the longest subsequence common to all sequences in a set of sequences (often just two). Note that a subsequence is different from a substring; with the former there is no need to be consecutive for the elements in the original sequence.

Let P and Q denote two ensemble pathlines with length m and n, and let $BlockIdx(P,i)$ be the index of the block where the $i-th$ point in P locates. The recursive definition of our LCSS distance is given by Equation (2).

$$L(i,j) = \begin{cases} 0, & if\ i=0\ or\ j=0, \\ 1+L(i-1,j-1), & if\ i,j>0\ and \\ & BlockIdx(P,i) = BlockIdx(Q,j), \\ max(L(i-1,j),L(i,j-1)) & if\ i,j>0\ and \\ & BlockIdx(P,i) \neq BlockIdx(Q,j). \end{cases} \quad (2)$$

To encode the ensemble pathlines into LCSS sequences (the input series of LCSS algorithm), we need to partition the data into multiple blocks. A unique sequence code will be encoded from each block index. Both the pointwise uncertainty computation [7] and the band depth computation [22] select every two pathlines each time, in order to compute the individual distance. Then the combinatorial groups of cases are summarized in to compute the overall distance and the inclusion probability, respectively. In our method, this scheme is followed to perform LCSS computation for every two pathlines each time. Then the overall distance is summed across all ensemble runs and normalized. If there are N different simulation runs in an ensemble, the normalized similarity value can be calculated according to Equation (3), where $LCSS(run_p, run_q)$ is the LCSS length (similarity) between any two ensemble pathlines (P and Q for run_p and run_q) at an identical seeding position.

$$S_{p,q} = \frac{2}{N(N-1)}\left(\sum_{p=1}^{N-1}\sum_{q=p+1}^{N}\frac{LCSS(run_p, run_q)}{max(m,n)}\right) \quad (3)$$

Figure 3 depicts a 2D illustration of block-index encoding for two pathlines. The longest common subsequence for these two runs is (0, 1, 2, 2, 3, 6, 8, 9). We consider all timesteps of the pathline as the elements of the sequence. Therefore, the length of common subsequence in block 2 should be 2, and the length of the longest common subsequence is 8 instead of 7. Then the normalized similarity value is obtained by dividing the maximum timesteps of the pathlines according to Equation (3). The normalized similarity value is approximately 0.615 (8 out of 13) in this case. Likewise, the normalized variation value is 0.385 (5 out of 13).

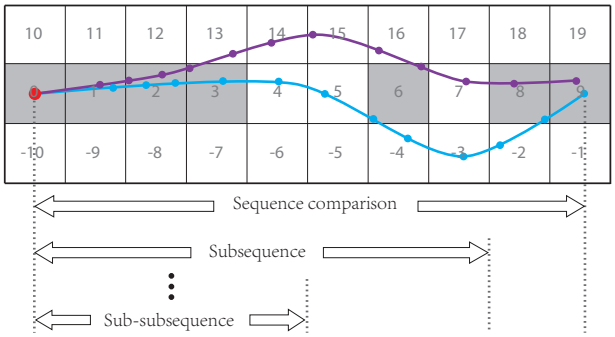

Figure 3: 2D illustration of the block-index encoding and subsequence comparison for multiscale temporal comparison. The block-index sequence of *Run 0* is (0, 1, 2, 2, 3, 13, 14, 15, 16, 6, 7, 8, 9). The sequence of *Run 1* is (0, 1, 2, 2, 3, 4, 5, 6, -4, -3, -2, 8, 9).

3.2 Variation Field and Similarity Field Generation

Domain scientists are interested in the regions with high variation and high similarity across different simulation runs. Thus, in this work we compute similarity and variation by using different block sizes to filter out the uninteresting in-between values. If scientists want to see the regions with high variation, the block sizes can be set at a relatively larger value. Likewise, if they want to see the regions with high similarity, the block size can be set at a smaller value. This scheme makes the distributions of both similarity values and variation values much more balanced. In contrast, using identical block size to compute the similarity field and variation field will often induce too much visual clutter; and the distributions of the final distance fields will often be imbalanced. Section 6.1 provides more details about the disadvantages of using identical block sizes.

Figure 4a shows two partitioning cases with different scales of block sizes. We can easily get the regions with high variation values by using a larger block size, and can obtain high similarity regions by using a smaller block size. Figure 4b shows the encoding metaphor for the 3D case, which indicates the process of assigning a block index to each block. If the LCSS block size is given, the raw data domain can be logically partitioned into multiple blocks around the seeding point, which is the original point of the encoding coordinates. After the variation field and similarity field are generated, we employ a clustering algorithm named DBSCAN [4] to highlight the regions with high variation. The clustering algorithm selection in the final visualization stage is not crucial; it is used just to draw a box to highlight the relatively continuous regions with high variation. We select DBSCAN because it can group points that are closely packed together, especially for the points with many nearby neighbors.

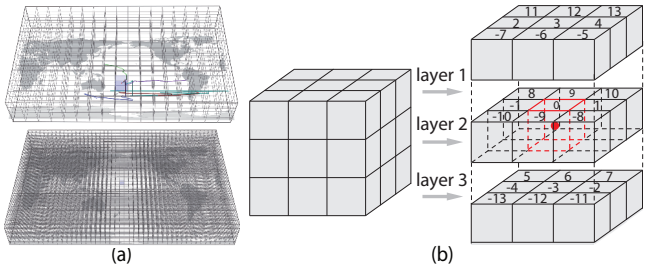

Figure 4: LCSS encoding metaphor: (a) two different spatial scales to partition the data; (b) Encode the block-index into LCSS sequence code along the seed point. The central red box shows the block containing the seed of the current ensemble pathlines.

3.3 Multiscale Temporal Comparison at Lower Storage Cost

Scientists often are interested in analyzing the ensemble behavior simultaneously over multiple time ranges. They even need to do temporal exploration in a multiscale way, for example, to analyze the data over one hour, one day, or one month. However, the computation of pathline is expensive because of the intensive integration. Pathlines can take 1,000 times more space than the unsteady flow field itself does [7]. Thus a multiscale temporal comparison for the traditional pointwise method is challenging. Moreover, both PCA-based variance and the FTLE require repeatedly tracing the pathlines for revealing the temporal trends, while our LCSS-based method requires only one tracing, because we can compute LCSS between sub-subsequences as shown in Figure 3. Furthermore, our method allows multiscale temporal comparison at a lower storage cost. We store the LCSS sequences that take much less space. Each resampled pathline P' in Equation (1) should be encoded into LCSS sequence codes $C = \{c_1, c_2, \cdots, c_n\}$, which is the input of the LCSS algorithm. Therefore, in theory, about one-third of the storage space is needed, because it needs only a single integer for an LCSS sequence code instead of three consecutive floating points for the intermediate pathlines (Equation (1)).

4 Implementation

We integrate the LCSS computation into a modified version of the DStep framework [12] to boost the efficiency of pathline tracing. It can achieve high scalability by its amphibious scheme on data-parallel and task-parallel, making the most intensive computation (i.e., pathline tracing) more efficient and effective. In the framework, processors are dynamically assigned to four types of roles: `steppers`, `reducers`, `writers`, and `communicators`. Two public callback functions, `dstep()` and `reduce()`, are provided to conduct the domain traversal across the whole ensemble field. Algorithm 1 is the pseudo code of the function `reduce()`. In this function, the ensemble pathlines first are resampled uniformly along the timestep; then their LCSS sequence code can be obtained by encoding the block index.

5 Evaluation

We test and evaluate our approach on three different datasets. The first one is a synthetic dataset. The second one is a simulation dataset, which is simulated by the Weather Research and Forecasting (WRF) Model.[1] The third one is also a simulation dataset, which was simulated by the Atmospheric General Circulation Model (AGCM) of Goddard Earth Observing System, Version 5 (GEOS-5). We conduct all our experiments in a parallel environment. The platform is a PC cluster with 8 nodes, each equipped with two Intel Xeon E5520 CPUs (quad core), operating at 2.26 GHz and with 48 GB main memory.

[1] http://www.wrf-model.org

Algorithm 1 `reduce()` function.

function REDUCE(seed, pathline_set[])
  for $p = 1$ to N **do** ▷ *Resample pathlines and compute their sequence codes*
    pathlines$_p$ ← resample_pathline(pathline_set$_p$)
    sequence_run$_p$ ← get_sequence_code(pathlines$_p$)
  end for
  for $p = 1$ to $N-1$ **do**
    for $q = p+1$ to N **do** ▷ *Compute LCSS between every two runs*
      $S_{p,q}$(seed) ← LCSS(sequence_run$_p$, sequence_run$_q$)
    end for
  end for
  V(seed) ← $1.0 - \frac{2}{N(N-1)}(\sum_{p=1}^{N-1}\sum_{q=p+1}^{N} S_{p,q}(\text{seed}))$
  for $p = 1$ to N **do**
    emit_write(seed, sequence_run$_p$)
  end for
  save_variation(V(seed)) ▷ *Save variation value into a global array to write latter*
end function

5.1 Sensitivity to Outliers

The synthetic dataset has two runs, the base run is a cylinder flow dataset (Reynolds number is 100) on a 400×100 grid. The velocity magnitude of the base run is about 0.05 in average. The synthesis run is generated by adding Gaussian noise ($\mu = 0, \sigma^2 = 0.001$) to each velocity component of the base run. Figure 5 shows the ensemble pathlines traced from the synthetic data. We conduct the tests of sensitivity to outliers on the pointwise distance, the DTW distance, and the LCSS distance. For the DTW distance, we use an improved version of the DTW algorithm, named FastDTW [17], to perform all the comparison tests. The block-index encoding function of DTW is the same as that of LCSS; hence, the input of DTW and LCSS are identical.

In this experiment, we find the pointwise distance is more sensitive to outliers, compared with the other two elastic distances (i.e., DTW and LCSS). If we select 0.5% of the grid cells in the synthesis run and add random noise to their velocity components, there will be 97.6% grid cells varied by more than 1%, 96.98% grid cells varied by more than 5%, and 96.23% grid cells varied by more than 10% for the pointwise distance, as shown in Table 1 ("PW" is short for the pointwise distance hereafter). There are two reasons. First, the pointwise distance deals with time series data of equal length. It needs to truncate the longer one if their lengths are not exactly the same. Second, the pointwise distance may accumulate the differences produced by outliers along its life-time. It is not enough to use only one threshold to measure the sensitivity. Thus we use three thresholds to reveal the distribution of the changing rate. We notice that all distance fields are normalized into the ranges between 0.0 and 1.0. From Table 1, we can also see that DTW is a little more sensitive to outliers than LCSS.

Distance	Sensitivity to Outliers		
	1%	5%	10%
PW	97.60%	96.98%	96.23%
DTW	59.11%	58.72%	58.22%
LCSS	55.03%	54.64%	52.59%

Table 1: Sensitivity to outliers when the noise is 0.5%. All distance fields are normalized. The sensitivity are measured by three changing rate thresholds.

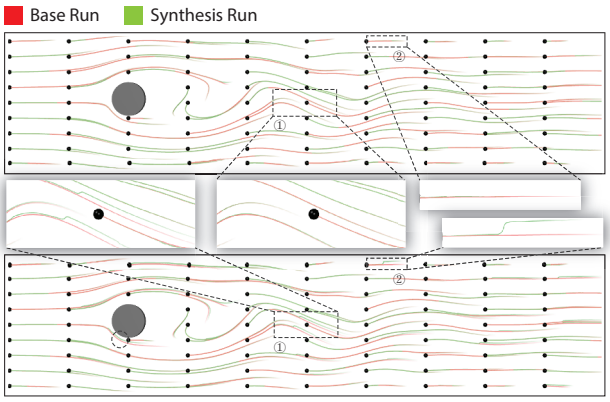

Figure 5: Pathlines traced in the 2D synthetic ensemble data. Top: pathlines of two runs (base run and original synthesis run). Bottom: additional 0.5% noise and data missing (velocity component is 0) are randomly added to each velocity component of the synthesis run.

5.2 Sensitivity to Pathline Timestep

If the pathlines are traced by a unreasonable timestep or are down sampled, the DTW distance function will be affected more severely because it is more sensitive to the timestep. In this section, we conduct a sensitivity test on the timestep for the three distances. We resample the ensemble pathlines by changing the timestep from 0.2 to 0.5, 1.0, and 1.5, as shown in Table 2. We expect that the pointwise distance is the least sensitive to the sampling timestep, because the Euclidean distance would be roughly unchanged after distance field normalization. The truth is that the pointwise distance values in the vector field will be scaled by the sampling rates accordingly. However, DTW is much more sensitive than LCSS, because all the elements in DTW must be matched, as shown in Figure 1c. Furthermore, the beginning and end timesteps in one series are always mapped to the beginning and end timesteps of the other series [13].

Timestep	Distance	Sensitivity to Timestep		
		5%	15%	30%
0.5	PW	5.82%	2.39%	1.28%
	DTW	57.40%	44.96%	32.96%
	LCSS	49.43%	26.44%	13.49%
1.0	PW	10.28%	4.04%	2.16%
	DTW	64.05%	53.66%	45.13%
	LCSS	56.41%	31.91%	17.42%
1.5	PW	15.56%	5.77%	3.18%
	DTW	69.37%	60.43%	53.43%
	LCSS	56.12%	32.93%	19.58%

Table 2: Sensitivity of the three distances to resampling timestep on WRF data. The baseline timestep is 0.2, and the resampling timesteps are 0.5, 1.0, and 1.5.

5.3 Ground Truth Evaluation

To further evaluate our LCSS-based distance measurement compared with pointwise distance and DTW distance approaches, we perform a ground truth test on a 2D cylinder flow dataset. A new synthesis run (different from the above-mentioned synthesis run in the outlier sensitivity test) is generated by adding noise with random directions. From the bottom to the top of the data domain, the intensity of the random noise is increasing: 0.0% to 0.05%, which is much less than that of the above outlier sensitivity test.

All three distance measurements are applied to compute the variation. The results are shown in Figure 6. We find that the LCSS-based distance works well in revealing the ground truth (i.e., the gradual changes from high to low), while the pointwise variation field and DTW variation field show strong discontinuous, which

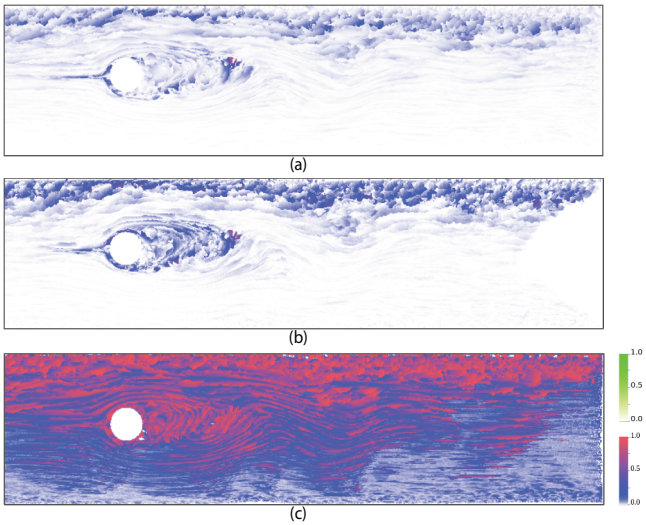

Figure 6: Ground truth test on a 2D synthetic time-varying flow dataset: (a) variation field produced by pointwise distance; (b) variation field produced by DTW distance; (c) variation field produced by our LCSS distance.

hide useful information because of the lack of enough accuracy. Furthermore, we find about 80% pointwise variations distributed among the range from 0.00 to 0.01 (within 1%), which greatly degenerates the final visualization. Throughout this paper, we employ a blue-red color scale for the variation field (red is higher) and a yellow-green color scale for the similarity field (green is higher).

5.4 Performance and Storage Cost

We test the performance of the three distance measurements on synthetic data and simulation data. Three different step sizes are taken to compute the average timing results. Each result is the average performance over three tests. As shown in Table 3, the performance of LCSS is a little better than that of DTW on both the synthetic and the simulation data especially for short sequences. But when the length of sequences is relatively longer, their timing results are closer. Although the performance of the pointwise distance is the best (it just summarizes the pointwise Euclidean distances), it has too many drawbacks, as claimed in this paper.

Dataset	Step size	Length	PW Time	DTW Time	LCSS Time
Synthetic Data	0.01	242.85	0.16	19.45	14.89
	0.02	121.45	0.085	9.71	3.82
	0.05	48.43	0.038	3.83	0.64
WRF	0.05	166.09	0.69	125.38	85.42
	0.10	83.56	0.37	64.25	21.50
	0.25	33.77	0.18	27.80	3.53

Table 3: Average timing results (in seconds) for the three distances computations. The numbers of processors of the two tests are 2 and 32, respectively. The total numbers of seeds are 80,000 and 540,000, respectively. "Length" is the average length of the traced pathline. The "PW Time," "DTW Time," and the "LCSS Time" are the corresponding distance function call times on all nodes.

Our method allows multi-temporal comparison without repeatedly tracing pathlines, because it enables sub-subsequence comparison with different lengths. More important, it allows reuse of the LCSS sequence codes (block indices) to do multiscale temporal comparison under lower storage cost; because we store the integer sequence codes, which takes much less storage than the pathlines themselves do. Table 4 shows the comparative storage cost for the intermediate pathlines and the LCSS sequence codes. For the WRF data and the GEOS-5 data, the storage cost for the sequence codes

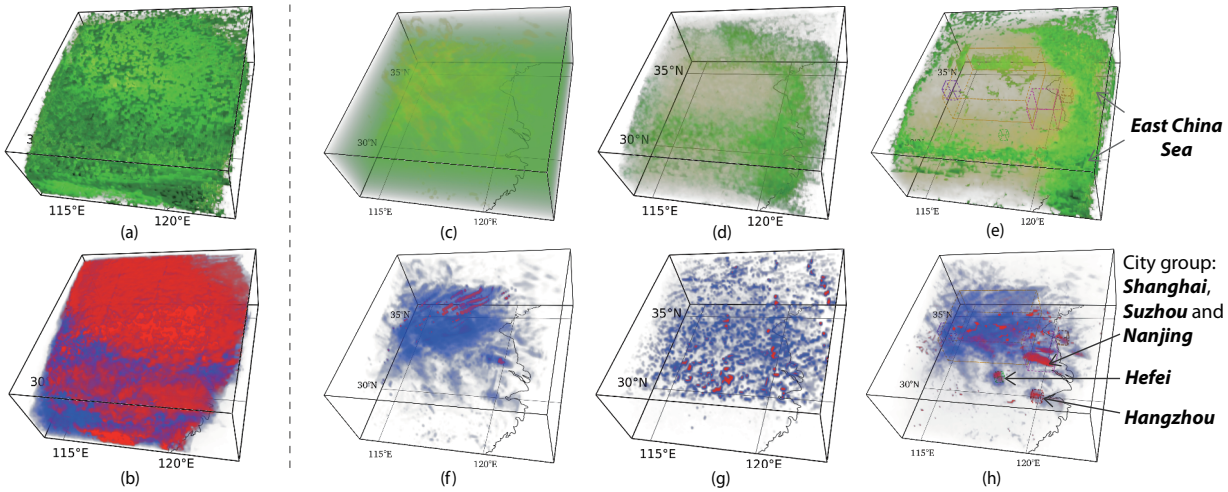

Figure 7: (a) Similarity field output by LCSS method with the same block size as (h); (b) variation field output by LCSS method with the same block size as (e); similarity field (c) and variation field (f) output by pointwise method; similarity field (d) and variation field (g) output by DTW method; similarity field (e) and variation field (h) output by LCSS method; The boxes in (e) and (h) are clustering regions with high LCSS variation.

are, respectively, 45.52% and 29.47% of that for pathlines themselves, which is significant when one wishes to make frequent multiscale temporal comparisons.

Dataset	S_{po}	S_{lo}	R_o
WRF	18.43 GB	8.39 GB	45.52%
GEOS-5	48.38 GB	14.26 GB	29.47%

Table 4: Storage cost between the intermediate pathlines and the LCSS sequence code. S_{po} and S_{lo} are the storage costs of pathlines and LCSS sequence code, respectively; R_o is the storage percentage of the original LCSS sequece codes in the original pathlines.

6 Results

In this section, we compare the results generated by our method and the two comparison methods (i.e., pointwise and DTW) based on two simulation datasets.

6.1 WRF Simulation Data

WRF simulations often consist of two runs. One run is a base run, which is derived from the real observation data. The other run is a comparison run, which is often simulated under some idealized conditions. With large-scale industrialization and increasing urbanization all over the world, scientists predict urbanization as one of the key factors in weather climate change. In this data, the simulation is conducted spatially on eastern China, which includes many metropolitans such as Shanghai, Nanjing, Hangzhou, Hefei, etc. They are China's most urbanized and industrialized cities.

The idealized condition of this WRF data is that the urban areas are replaced by vegetation landuse. The scientists who conducted this simulation wanted to determine the impact of urbanization on climate change in eastern China. The highly urbanized regions are expected to have high variation according to the input parameters of the simulation. The data is organized in grid cells $100 \times 100 \times 27$. The time range of this simulation is from 7/1/2012 00:00:00 UTC to 7/10/2012 18:00:00 UTC. Hourly average data is generated by this simulation, and the total storage size is about 8 GB.

If we use the same block size to compute the similarity and variation, it will often produces too much visual clutter, as shown in the left part of Figure 7. Different block sizes can help filter out uninteresting values. Therefore, we set the block size for the variation field larger than that for the similarity field. After many tests, we found $16\times$ and $1\times$ step sizes to be an optimal pair.

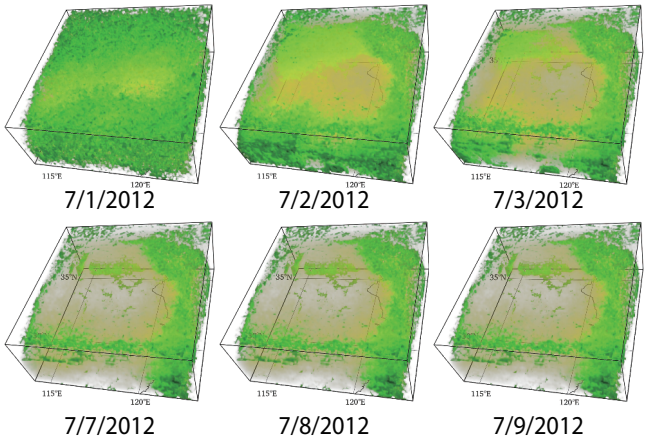

Figure 8: Multiscale temporal comparison for WRF data. Daily comparative visualization from 7/1/2012 to 7/9/2012. Daily trends (detail) and weekly trends (overview) can be analyzed simultaneously.

The right part of Figure 7 shows the comparative results. We see that the similarity field (Figure 7c) computed by the pointwise method is sensitive to outliers and that the distribution of the similarity values is skewed. Thus, it is difficult to get a proper transfer function to reveal detailed information even after normalization. Compared with the pointwise similarity field, the similarity field (Figure 7d) output by the DTW distance is more expressive. However, it still has many vague regions between mainland China and the East China Sea. In our method, instead, the similarity field reveals more meaningful and interesting results. For example, the similarities are high over the East China Sea and the regions with high altitude, where the influences of urbanization are relatively low (domain knowledge from the experts), as shown in Figure 7e.

Our LCSS variation field can also reveal much more expressive results. The variations over the primary metropolitans in eastern China are quite large. For example, three clustered regions are highlighted in the variation field. The first region is over the city group including Shanghai (121.4°E, 31.2°N), Suzhou (120.6°E, 31.3°N), Wuxi (120.6°E, 31.3°N), Changzhou (120.3°E, 31.6°N), and Nanjing (118.7°E, 119.9°N) in the pink box in Figure 7h. The second region is over Hefei (117.2°E, 31.5°N), the provincial capital in Anhui province as shown in the green box. The third region is over Hangzhou (120.2°E, 30.3°N), the provincial capital of Zhejiang province, as shown in the red box. However, we cannot find

these interesting results in the pointwise variation field (Figure 7f) and DTW variation field (Figure 7g), although the overall distribution in the pointwise variation field is roughly similar to that of the LCSS variation field (Figure 7h).

Figure 8 illustrates a case for multi-temporal comparison. The similarity field is sharply diversified on the second day (7/2/2012 00:00:00 UTC). One week later, it is becoming more and more steady, and is almost invariable from 7/7/2012 00:00:00 UTC to 7/9/2012 00:00:00 UTC. Overall, the spatial distribution of the similarity is approximately fixed from the second day and gets steadier from then on.

We notice that our method is not sensitive to the block size. We have changed the block size by 10%, 20%, and 50% and find similar results, because the block size is used just to filter out uninteresting data. Moreover, although we use different block sizes, only one pass computation is needed in our parallel pathline tracer.

6.2 GEOS-5 Simulation Data

We also use the GEOS-5 simulation data to test our approach. The data was generated by the simulation model with 72 pressure levels and a horizontal resolution of 1° latitude $\times 1.25^\circ$ longitude. It covers the troposphere, stratosphere, and partial mesosphere (upper bound is 85 km). It has 8 runs with standard monthly output from January, 2000 to December, 2001. Each run of the data is stored in 24 individual files corresponding to different timesteps. The total size of this simulation data is about 76 GB.

We find that the results output by LCSS distance on GEOS-5 data are more expressive than that of pointwise distance. We do not get DTW distance results for the GEOS-5 data because the computation on DTW is too costly; the resources of our parallel environment will quickly run out in the `reduce()` function. From the variation field, we can see that the variation near the North Pole and the the South Pole is extremely high compared with most of other regions, especially for the regions around the equator. Additionally, we can see the high similarity distributed in low-altitude and high-altitude areas around the stratosphere, as shown in Figure 9d. This finding is consistent with the domain knowledge provided by domain experts and cannot be found with the pointwise method in Figure 9a.

More important, we can see that the variation field computed by our LCSS-based method (Figure 9b) is different from that computed by the pointwise method (Figure 9a). Most of the regions with high variation computed by the pointwise method are near the equator. However, the ground truth is that the ensemble pathlines behave similarly around the equator, as shown in Figure 9c. There are two major reasons why the pointwise distance on this data is different. First, the pointwise distance deals with time series data of equal length; it needs to truncate the longer one if their lengths are not the same. Second, the pointwise distance is more sensitive to outliers, missing data, and extreme values due to severe misalignment. We observe that the velocity around the equator is usually much larger and has a higher dynamic range of distribution compared with that of other regions (e.g., the North Pole Circle and the South Pole Circle), which will subsequently result in much more extreme values and outliers. Moreover, considerable data is missing around the equator during the data collection and processing of the simulation.

7 Domain Expert Feedback

We have consulted domain experts in climate and environmental science and received considerable positive feedback. The scientist who provides us the WRF data has been engaged in climate simulations for a long time. He showed much appreciation about our approach and was glad to see our findings. Feedback about the WRF data tests is listed as follows.

(1) *The high variation regions over the primary metropolitans in east China, and the high similarity regions over the East China Sea, are consistent with our knowledge and our initial expectations.*

(2) *The diversity of similarity that goes steady in the first three days is a significant information about this WRF simulation.*

(3) *It would be better if it can provide enough analysis on how diversity behaves between different runs, because the diversified velocity can help us explore the diffusion pattern.*

We have also received feedback from the domain expert about the GEOS-5 data.

(1) *Many high similarity distributed in the stratosphere is consistent with our knowledge.*

(2) *Ensemble pathlines around the equator coincide with the findings.*

(3) *The much less storage requirement of the proposed approach is quite significant for the ensemble simulation analysis.*

According to the feedback, our proposed approach is capable of achieving more meaningful and expressive results than conventional methods achieve. The reason is that the LCSS distance is more robust and more flexible for computing the distance among ensemble pathlines. They can provide more accurate visualization results.

8 Discussion

In our experiments, we find that our method is relatively robust to block size. If we change the block size slightly, the distributions of both similarity fields and variation fields are almost unchanged. In the WRF simulation data, we increase the two block sizes by 10%, 20%, 50%, then decrease them by 10%, 20%, 50%, and find distributions similar to that of Figure 7e and Figure 7h. That is, we can also find the same results as described in Section 6. The primary role of block size is to filter the distance values. We use a smaller block size to compute the similarity field and a larger block size to compute the variation field in order to get the regions with high similarity and high variation, respectively. Even if the distribution is changed slightly, we can adjust the global transfer function to a small extent to get the same results. Therefore, we can get the same results as long as our method satisfies two conditions. First, it needs a smaller block size for similarity computation than that of variation computation. Second, the block size should not be extremely large or extremely small. Both conditions are easy to satisfy.

In our results, we show just the overall similarity field and variation field. However, one can easily compute results for any two vector fields among the ensembles, because the LCSS algorithm is designed to compute the distance between two vector fields. The overall results are computed from each pair of two vector fields according to Equation (3), for two reasons. First, each pair of vector fields are potential to be used in further analysis. Second, the LCSS algorithm based on multiple sequence is inefficient.

Our method still has some limitations, however. First, we need to estimate the average stepsize in order to assign the block size in advance. If the block size is extremely large or extremely small, our approach will fail. However, one can easily estimate the step size; one just needs to write an applet to estimate the average step size of the vector field for each run. Second, converting pathlines into sequences can result in lossing information. We store the LCSS sequence codes to support reuse because the pathline tracing is time-consuming. There are three components (x, y, and z) for each point on the original pathline when the data is a 3D vector field, but only one component (LCSS sequence code) for each point in our storage. However, it is still useful unless the user wants to compute the LCSS distances by different sampling rates. Third, the current version of the implementation does not support vector fields with a nonuniform grid, (i.e., unstructured grid, hybrid grid, etc.), because we use a classic linear 4D interpolation based on uniform grid when tracing the ensemble pathlines. Nevertheless, our approach can be used in nonuniform grid cases as long as the interpolation and pathline tracing are extended to those data.

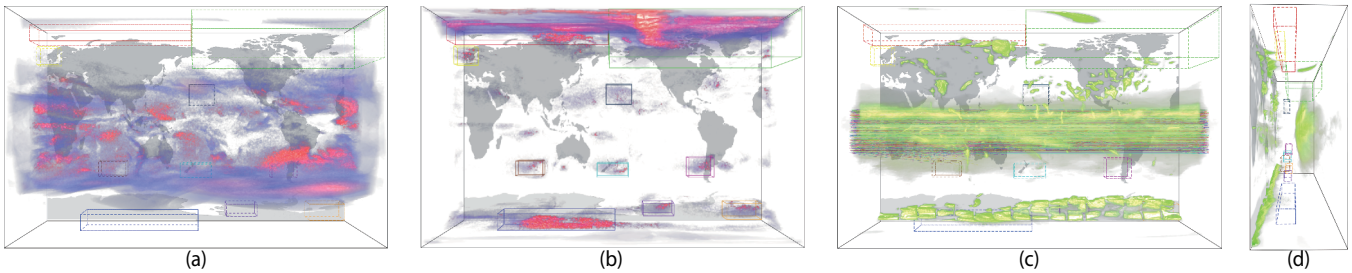

Figure 9: Comparative results for GEOS-5 data. (a) Variation field computed by pointwise method. (b) Variation field (b) and similarity field (c) computed by LCSS distance. The LCSS similarity field and the ensemble pathlines around the equator behave similarly. (d) Similarity field results from another perspective to see the similarity distribution along altitude. The boxes in (b) and dashed boxes in (a), (c), and (d) represent the clustered regions with high LCSS variation.

9 Conclusions and Future Work

We propose a novel LCSS-based measurement to compute the distance among vector field ensembles. Compared with the traditional methods, our approach is robust to outliers, missing data, and the pathline timesteps. It enables to reveal temporal trends at much less storage cost. We evaluated our method on both synthetic and simulation data, demonstrating the robustness of the proposed approach. We also consulted domain experts to evaluate the final visualization results. The feedback indicates that the proposed approach can generate more meaningful and expressive results, because the results are well matched to the domain knowledge.

We plan to extend our approach to analyze scalar features. We also plan to provide more visual analysis on how diversity behaves between different simulation runs.

Acknowledgements

We would like to thank Dr. Junfeng Liu and Dr. Xiaoguang Ma for their valuable comments and suggestions during the development of this work. This work is supported by NSFC No. 61170204, NSFC Key Project No. 61232012, and the "Strategic Priority Research Program - Climate Change: Carbon Budget and Relevant Issues" of the Chinese Academy of Sciences Grant No. XDA05040205. This material is also partially based upon work supported by the U.S. Department of Energy, Office of Science, under contract number DE-AC02-06CH11357.

References

[1] E. W. Chambers and Y. Wang. Measuring similarity between curves on 2-manifolds via homotopy area. In *Proceedings of the Twenty-ninth Annual Symposium on Computational Geometry*, SoCG '13, pages 425–434, New York, NY, USA, 2013. ACM.

[2] Y. Chen, J. D. Cohen, and J. H. Krolik. Similarity-guided streamline placement with error evaluation. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1448–1455, 2007.

[3] I. Demir, C. Dick, and R. Westermann. Multi-charts for comparative 3d ensemble visualization. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2694–2703, 2014.

[4] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining*, pages 226–231. AAAI Press, 1996.

[5] F. Ferstl, K. Brger, and R. Westermann. Streamline variability plots for characterizing the uncertainty in vector field ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):767–776, 2016.

[6] L. Gosink, K. Bensema, T. Pulsipher, H. Obermaier, M. Henry, H. Childs, and K. Joy. Characterizing and visualizing predictive uncertainty in numerical ensembles through Bayesian model averaging. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2703–2712, 2013.

[7] H. Guo, X. Yuan, J. Huang, and X. Zhu. Coupled ensemble flow line advection and analysis. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2733–2742, 2013.

[8] G. Haller. Distinguished material surfaces and coherent structures in three-dimensional fluid flows. *Physica D: Nonlinear Phenomena*, 149(4):248–277, 2001.

[9] D. S. Hirschberg. A linear space algorithm for computing maximal common subsequences. *Communications of the ACM*, 18(6):341–343, 1975.

[10] M. Hummel, H. Obermaier, C. Garth, and K. I. Joy. Comparative visual analysis of Lagrangian transport in CFD ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2743–2752, 2013.

[11] M. Jarema, I. Demir, J. Kehrer, and R. Westermann. Comparative visual analysis of vector field ensembles. In *IEEE Visual Analytics Science and Technology*, pages 81–88, 2015.

[12] W. Kendall, J. Wang, M. Allen, T. Peterka, J. Huang, and D. Erickson. Simplified parallel domain traversal. In *Proceedings of the ACM Conference on Supercomputing*, pages 1–10, 2011.

[13] T.-Y. Lee and H.-W. Shen. Visualization and exploration of temporal trend relationships in multivariate time-varying data. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1359–1366, 2009.

[14] R. Liu, H. Guo, and X. Yuan. Seismic structure extraction based on multi-scale sensitivity analysis. *Journal of Visualization*, 17(3):157–166, 2014.

[15] R. Liu, H. Guo, and X. Yuan. A bottom-up scheme for user-defined feature comparison in ensemble data. In *ACM SIGGRAPH Asia 2015 Symposium on Visualization in High Performance Computing*, pages 1–4, 2015.

[16] 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(12):2654–2663, 2014.

[17] S. Salvador and P. Chan. FastDTW: Toward accurate dynamic time warping in linear time and space. In *KDD Workshop on Mining Temporal and Sequential Data*, pages 70–80, 2004.

[18] J. Sanyal, S. Zhang, J. Dyer, A. Mercer, P. Amburnand, 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.

[19] Q. Shen, A. Pang, and S. Uselton. Data level comparison of wind tunnel and computational fluid dynamics data. In *Proceedings of the IEEE Visualization 98*, pages 415–418, 1998.

[20] R. Sisneros, J. Huang, G. Ostrouchov, S. Ahern, and B. D. Semeraro. Contrasting climate ensembles: A model-based visualization approach for analyzing extreme events. *Procedia Computer Science*, 18:2347–2356, 2013.

[21] M. Vlachos, G. Kollios, and D. Gunopulos. Discovering similar multidimensional trajectories. In *Proceedings of 18th International Conference on Data Engineering*, pages 673–684, 2002.

[22] 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(12):2713–2722, 2013.