Dual Space Analysis of Turbulent Combustion Particle Data

Jishang Wei* University of California, Davis
Hongfeng Yu† Sandia National Laboratories
Ray W. Grout‡ National Renewable Energy Laboratory
Jacqueline H. Chen§ Sandia National Laboratories
Kwan-Liu Ma¶ University of California, Davis

ABSTRACT

Current simulations of turbulent flames are instrumented with particles to capture the dynamic behavior of combustion in next-generation engines. Categorizing the set of many millions of particles, each of which is featured with a history of its movement positions and changing thermo-chemical states, helps understand the turbulence mechanism. We introduce a dual-space method to analyze such data, starting by clustering the time series curves in the phase space of the data, and then visualizing the corresponding trajectories of each cluster in the physical space. To cluster time series curves, we adopt a model-based clustering technique in a two-stage scheme. In the first stage, the characteristics of shape and relative position are particularly concerned in classifying the time series curves, and in the second stage, within each group of curves, clustering is further conducted based on how the curves change over time. In our work, we perform the model-based clustering in a semi-supervised manner. Users' domain knowledge is integrated through intuitive interaction tools to steer the clustering process. Our dual-space method has been used to analyze particle data in combustion simulations and can also be applied to other scientific simulations involving particle trajectory analysis work.

1 INTRODUCTION

Next-generation combustion engines will operate in the non-conventional, mixed-mode, and turbulent conditions. Combustion processes in these environments, combined with new physical and chemical fuel properties, result in complicated interactions that are poorly understood at a fundamental level. To obtain a better understanding, direct numerical simulation (DNS) is used to capture and describe the key turbulence-chemistry interactions. S3D [9], a massively parallel solver, has been developed at Sandia National Laboratories to solve the DNS governing equations originating from a Eulerian viewpoint.

Besides the Eulerian viewpoint, in fluid mechanics, the Lagrangian viewpoint is another commonly used method to describe a flow. The Eulerian specification of the flow field is a way of looking at fluid motion that focuses on specific locations in the space through which the fluid flows as time passes [5, 23]. The Lagrangian specification of the flow field is a way of looking at fluid motion where the observer follows an individual fluid parcel as it moves through space and time [5, 23]. The parcel evolves along a path with the instantaneous position $\vec{x}(\vec{x}_0,t)$ and the initial position $\vec{x}_0$ according to [34]:

$$\frac{\partial \vec{x}(\vec{x}_0,t)}{\partial t} = \vec{u}(\vec{x}_0,t). \quad (1)$$

*e-mail:jswei@ucdavis.edu
†e-mail:hyu@sandia.gov
‡e-mail:Ray.Grout@nrel.gov
§e-mail:jhchen@sandia.gov
¶e-mail:ma@cs.ucdavis.edu

IEEE Pacific Visualisation Symposium 2011
1 - 4 March, Hong Kong, China
978-1-61284-934-8/11/$26.00 ©2011 IEEE

where $\vec{u}(\vec{x}_0,t)$ is the instantaneous field velocity at the position $\vec{x}(\vec{x}_0,t)$. Since the transport of combustion turbulence is dominated in an advective way, the Lagrangian description is natural and useful for the analysis of combustion turbulence [34]. Recent combustion simulations of a turbulent lifted autoignitive ethylene/air jet flame in a hot air coflow are instrumented with particles originating from both the fuel and oxidizer sources. These simulations provide Lagrangian description of the combustion environment. The passive tracer particles are disseminated in the combustion flames and advected by the velocity field *in situ* with a fourth order Runge-Kutta time advance. At each Runge-Kutta substep, trilinear interpolation is used to determine the particle velocity from the Eulerian solution. While the particle position is integrated during the simulation time, the thermo-chemical state (temperature, composition, etc.), interpolated from the Eulerian grid to the particle positions, is also saved. In this sense, DNS provides a set of particles, each of which contains a record of the history of its movement positions and changing thermo-chemical states.

In our work, we study the particles' movement trajectories in the physical space, and their thermo-chemical evolution, represented as time series curves, in the phase space. The phase space is a 2D domain with space dimensions of temperature and *mixture fraction*, two key combustion parameters encapsulating the unsteadiness associated with turbulent mixing and autoignition. The *mixture fraction* is a mixing measure of fuel and oxidizer, which takes a value of unity in pure fuel and zero in pure oxidizer. No matter in the physical space or in the phase space, with the passage of time, particles' histories can be recorded as a sequence of points. Thus, a trajectory or time series curve can be represented in the following manner,

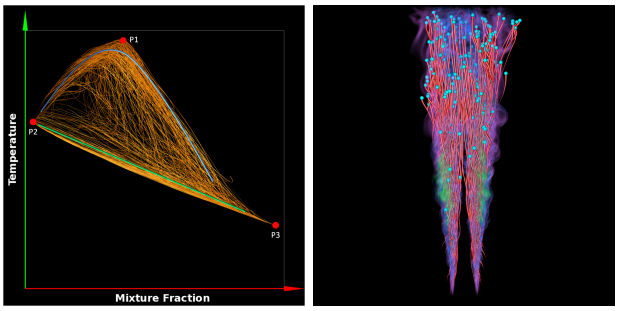
$$R = [\mathbf{r}_1, \mathbf{r}_2, ...\mathbf{r}_n] \quad (2)$$

where n, the number of sample timestamps in R, is defined as the length of R, and $\mathbf{r}_i$ is a data item of dimension d that is sampled at timestamp t_i.

Figure 1 (a) shows the particles' time series curves in the phase space, providing Lagrangian statistics correlating temperature and *mixture fraction*. Figure 1 (b) shows the corresponding trajectories of the same group of particles in the physical space. In Figure 1 (a), the green and red axes represent temperature and *mixture fraction* respectively. We can see that the path by which fluid particles (blue spots) traverse from the frozen flow mixing limit (bottom line in Figure 1 (a)) to the equilibrium limit (upper tent shaped line in Figure 1 (a)) is captured in the phase space of temperature and mixture fraction.

There are millions of particles in the simulations used to capture the dynamic behavior of combustion flames. Our work is motivated by the scientists' interests on what different patterns of time series curves are in the phase space and how particles' time series curves and trajectories are related. However, categorizing and understanding the large amount of particle histories is non-trivial. As shown in Figure 1, substantial clutter is introduced by heavily interweaving of dense lines, which brings unique challenges to scientists to perceive detailed correlation patterns.

This paper focuses on analyzing dynamic combustion behavior by categorizing particles' time series curves in the phase space and



(a) (b)

Figure 1: (a) shows the time series curves representing correlation between temperature and *mixture fraction*, which are two key parameters in the combustion simulation of a turbulent lifted autoignitive ethylene/air jet flame. There is a great deal of clutter and it is hard to perceive detailed correlation patterns. (b) shows the corresponding particle trajectories in the physical space, with volume rendering of the hydroperoxy field.

visualizing particles' trajectories in the physical space. The relationship between the time series curves and trajectories is analyzed. The structure of this paper is organized as follows: right after a detailed introduction of the related work in Section 2, Section 3 talks about the dual-space analytical methodology we use, including automatic and interactive model-based clustering and two-stage cluster analysis; Section 4 shows the dual-space method we use to analyze the particle data; Section 5 summarizes the current work and discusses further research.

2 Related Work

How to visually analyze time-varying data has been broadly studied by the visualization researchers. In medical data analysis, Fang et al. [12] regarded a time-varying data set as a 3D array where each voxel contains a time-activity curve (TAC). He defined three similarity metrics to quantify the difference among TACs so that different regions of interest containing these TACs can be segmented and visualized. Van Wijk and Selow [30] proposed a cluster and calendar based analytical tool to explore and visualize univariate time series data. They utilize automatic hierarchical clustering to find similar daily patterns for the analysis of time series data of one year. The results are visualized using two conventional representations: average daily patterns of clusters are shown as graphs, and the days per cluster are shown on a calendar.

Aside from automatic methods, interactive analysis approaches are also developed to extract interesting patterns in the time-varying data. For example, TimeSearcher [19] is such a time series exploratory and visualization tool that allows users to retrieve time series by creating queries. This is achieved by use of "TimeBoxes", which are rectangular query locators that specify the regions in which the users are interested within any given time series. Akiba and Ma [2] introduced a novel Tri-space visualization interface to address the problem of examining multivariate time-varying data. Konyha et al. [22] presented a valuable tool by combining the established visualization techniques, linked views, and advanced brushing features for interactive visual exploration and analysis of families of function graphs. To deal with overdrawing and visual cluttering when depicting large amounts of function graphs, Muigg et al. [26] developed a four-level focus+context interactive visualization method, with the context information for orientation and also three different levels of focus in every attribute view.

In practice, both automatic and interactive approaches have their own advantages and limitations. There has been a trend towards integrating the automatic methods and users' interaction in large complex data analysis and visualization [7, 20]. Woodring and Shen [32] proposed a technique to semi-automatically generate transfer functions for time-varying data via temporal clustering and sequencing. Lee and Shen [24] presented a new algorithm identifying important trend relationships among the variables based on how the values of the variables change over time and how those changes are related to each other in different spatial regions and time intervals. Schreck et al. [28] proposed a user-supervised SOM clustering algorithm that enables users to control and monitor the computation process visually to leverage their domain knowledge.

Our work follows the idea of integrating automatic data analysis with human domain knowledge, relying on interaction means. The automatic data analysis technique, namely, time series curves clustering in our work, has been a research focus in both the visualization and data mining communities and great advances have been made. To get a better understanding, interested readers are referred to the comprehensive surveys on visual analysis of time-oriented data [1] and time-series data mining [31, 21]. In general, regarding clustering time series data, similarity-based methods and model-based methods are the two major kinds. In similarity-based methods, it is critical to determine how to present the time series data and how to measure the similarity or distance between a pair of data objects. The commonly used distance metrics include Euclidean distance and dynamic time warping (DTW) distance [6]. For other complex data types, defining a good similarity measure is mostly data dependent and often requires expert domain knowledge. Ding et al. [11] conducted an extensive experimental consolidation on the state-of-the-art representation methods and similarity measures for time series data. In order to provide a comprehensive validation, the authors re-implemented 8 different representation methods and 9 similarity measures and their variants, and tested their effectiveness on 38 time series data sets from a wide variety of application domains.

Unlike similarity-based methods that assume some weak structure of the data, model-based methods assume some strong structure, the model. The commonly used models for time series data include [33]: markov chains, hidden markov models, regression models, and autoregressive moving average models. This category of clustering methods is often based on the assumption that the data are generated by a mixture of underlying probability distributions. The probabilistic representation allows for the derivation of consistent expectation-maximization (EM) learning algorithms for the clustering problem, in a sense of maximum likelihood (ML) estimation. Model-based clustering can handle complex time series data, such as those with different lengths.

3 Dual-Space Analysis Methodology

3.1 An Overview

Our method operates in two spaces, the phase space and the physical space, to analyze and visualize particle data. The phase space is composed of the particles' attributes parameters, specifically, temperature and mixture fraction. In our study, the physical space is the 3D simulation domain, in which the particles are advected. In our system, we first automatically cluster time series curves based on their shapes and positions in the phase space. Next, according to the preliminary clustering results with much less clutter, users can set the clustering parameters through interaction tools and reconduct cluster analysis to refine the results. Last, the movement of corresponding particles in each cluster is visualized to verify the users' hypothesis.

Automatic cluster analysis is a critical component in our system. As we introduced in Chapter 2, there are two major groups of clustering algorighms: similarity-based methods and model-based methods. A hierarchical similarity-based method seeks to build a hierarchy of clusters, starting from either each individual data ojbect as a cluster (agglomerative clustering) or the whole collec-

tion of data as a cluster (divisive clustering). The hierarchical technique creates a set of nested clusterings, making a good visualization. However, since pairwise distances have to be computed, hierarchical methods tend to have computational complexity that is quadratic in the number of data patterns and hence becomes prohibitive for large data sets like in our case. A partitional similarity-based method, such as the K-means algorithm, partitions data objects into a number (often specified by a priori) of clusters directly according to some optimization criterion. This kind of methods have computational complexity that is linear to the number of data size. Nevertheless, regarding time series or trajectory data, especially when the lengths of which vary, it is nontrivial to design an apropriate similarity metrics. And so is the case for hierarchical similarity-based methods. Taking these situations into consideration, we adopt a model-based clustering method, which assumes a strong statistical model about the data. This method provides a principled approach for handling the problem of modeling and clustering time series of different lengths and can incorporate prior knowledge naturally. The only problem with model-based clustering is that it reqiures users' setting how many clusters the data should be partitioned into. Since we build an interactive environment for the users to examine data and determine the number of clusters, the model-based method fits well in our application.

The automatic clustering methods help disclose data patterns and facilitate the following visualization. However, automatic cluster analysis may not always generate satisfying results to fulfill the users requirements. As is stated in [17], knowledge discovery would be most effective if one could develop an environment for human-centered, exploratory mining of data, that is, where the human user is allowed to play a key role in the process. We adopt a similar idea as the semi-supervised clustering method in [4] by coupling both the users' domain knowledge and clustering algorithm through intuitive interaction methods. The user can intervene and guide the clustering process by setting the critical parameters in the clustering process, specifically, the number of clusters and the corresponding parameter values.

After categorizing particles' time series curves in the phase space by the clustering method, we incorporate line rendering and volume rendering to visualize the trajectories with respect to each cluster in the physical space. The trajectories are embedded into the surrounding instantaneous field data at each time step, and the particle movement can be animated simultaneously in the phase and physical spaces. By this means, the movement pattern of each particle category can be examined individually to reveal the evolution of the particles and the correlation between the particle attributes.

3.2 Automatic Clustering of Time Series Curves

3.2.1 Fitting Time Series Curves with B-spline

As with most problems in computer science, a suitable choice of data representation would lead to the ease and efficiency of time series curves clustering. In our application, since the domain experts discriminate time series curves according to their shapes and relative positions in the phase space, we are prone to use a curve representation method that can encompass both characteristics. The B-spline model matches our need. To represent a time series curve, we first fit it with a B-spline model and then sample a sequence of points at equal arc length along the modeled curve.

Uniform B-Spline A uniform B-spline is a convenient form to represent complex, smooth curves. It is in general chosen because of the ease of manipulation. An open cubic uniform B-spline can be written in a matrix notation as,

$$p(s) = [1 \;\; s \;\; s^2 \;\; s^3] \times \mathbf{B} \times \mathbf{G} \times \mathbf{p} \tag{3}$$

where $i \le s < i+1$ and

$$\mathbf{B} = \begin{bmatrix} -\frac{1}{6}i^3 & \frac{1}{6}(3i^3+3i^2-3i+1) & -\frac{1}{2}i^3-i^2+\frac{2}{3} & \frac{1}{2}(i+1)^3 \\ \frac{1}{2}i^2 & -\frac{1}{2}(3i-1)(i+1) & \frac{1}{2}(3i^2+4i) & -\frac{1}{2}(i+1)^2 \\ \frac{1}{2}i & \frac{1}{2}(3i+1) & -\frac{1}{2}(3i+2) & \frac{1}{2}(i+1) \\ \frac{1}{6} & -\frac{1}{6} & \frac{1}{2} & -\frac{1}{6} \end{bmatrix}$$

for the matrix $\mathbf{G}$,

$$G(m,\, n) = \begin{cases} 1 & \text{if } n = i+m-3 \\ 0 & \text{otherwise} \end{cases}$$

and $\mathbf{p}$ is the vector of control points,

$$\mathbf{p} = \begin{bmatrix} p_1 \\ \vdots \\ p_n \end{bmatrix}$$

We denote the B-spline basis matrix as $\mathbf{M}(s)$,

$$\mathbf{M}(s) \;=\; [1 \;\; s \;\; s^2 \;\; s^3] \times \mathbf{B} \times \mathbf{G} \tag{4}$$

which is a $1 \times n$ matrix, then Equation (3) can be written as,

$$p(s) = \mathbf{M}(s) \times \mathbf{p} \tag{5}$$

Arc Length Parameterization of B-spline Curves By being fitted to the uniform B-spline model, time series curves are defined as a function of parameter t in the range of $[t_{\min}, t_{\max}]$, where $t_{\min}$ and $t_{\max}$ correspond to the beginning and end of the curve. That is,

$$\mathbf{c}(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}.$$

In our application, the curve is represented by the sample points of equal arc length. Thus, it is desirable to evaluate parametric B-spline curves at points based on their arc length instead of the curve's original parameter. The curve needs to be represented as a function of parameter s in the range of $[0, L]$, where L is the total length of the curve. i.e.,

$$\mathbf{c}(s) = \begin{pmatrix} x(s) \\ y(s) \end{pmatrix}.$$

This parameterization operation can be implemented with numerical method, such as Runge-Kutta.

3.2.2 Cluster Analysis of Time Series Curves

B-spline Regression Mixture Models We model the time series curves $\mathbf{Y} = \{\mathbf{y}_1, \cdots, \mathbf{y}_n\}$ in the phase space with a B-spline regression model [14, 15, 16] in which arc length $\mathbf{s}_i$ is the independent variable. The regression equations can be defined as follows:

$$\mathbf{y}_i = \mathbf{M}_i\mathbf{p} + \varepsilon_i, \;\; \varepsilon_i \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}) \tag{6}$$

where $\mathbf{M}_i$ is the B-spline basis matrix, as in Equation (4), evaluated at $\mathbf{s}_i \;(\; \mathbf{s}_i = (s_1, \cdots, s_m)\;)$,

$$\mathbf{M}_i = [\mathbf{M}(s_1), \; \cdots, \; \mathbf{M}(s_m)]'$$

and $\mathbf{p}$ is control points vector. ε_i is a noise term following a Gaussian distribution, with the mean vector and covariance matrix as $\mathbf{0}$ and $\sigma^2\mathbf{I}$ respectively.

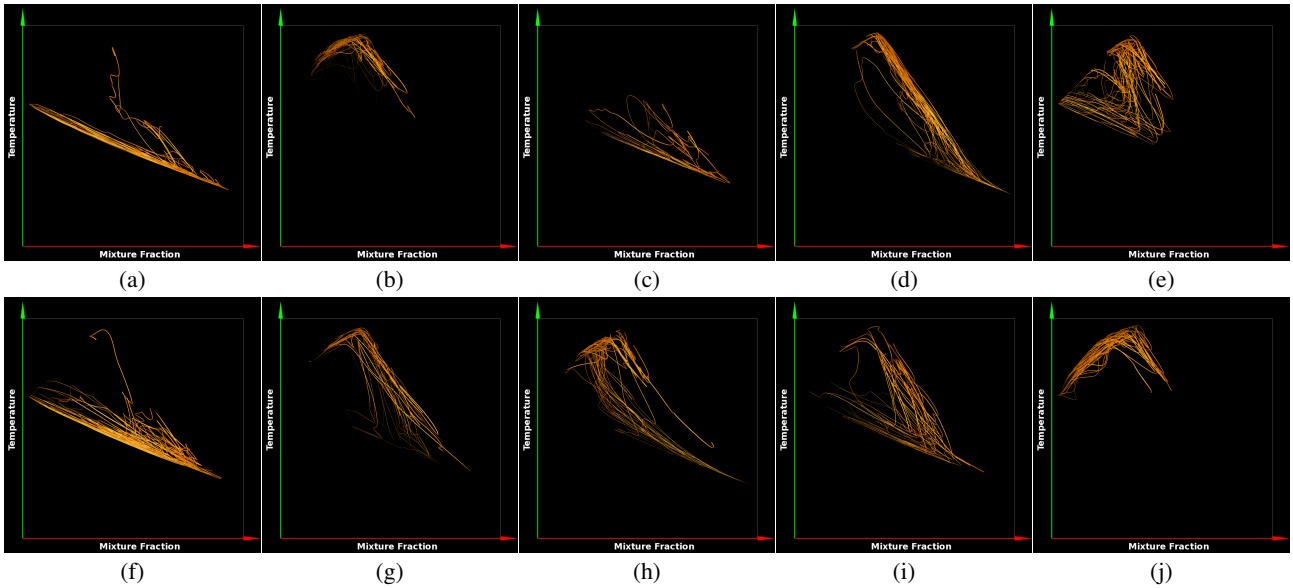


Figure 2: This figure shows the ten different groups of time series curves generated by the automatic model-based clustering algorithm. The result provides an initial partition of the curve with much less clutter, which can facilitate the user to adjust the clustering parameters and refine the clustering process with domain knowledge.

Model-based Clustering Model-based clustering can be regarded as the generalization of K-means algorithm [25, 18]. In the context of model-based clustering, the whole set of time series curves is assumed to derive from a mixture model of K components, which correspond to K clusters. Each component of the mixture model is associated with a probabilistic density function p_k. Then the mixture density for one curve $\mathbf{y}_i$ of length d is,

$$p(\mathbf{y}_i|\Theta) = \sum_k \alpha_k p_k(\mathbf{y}_i|\theta_k) \tag{7}$$

where α_k denotes the probability of assigning curve $\mathbf{y}_i$ to cluster k and θ_k is the distribution parameters of cluster k, with $\Theta = \{\theta_i, \cdots, \theta_K\}$. We use a normal distribution in our work. That means θ_k contains parameters of a d-dimensional mean vector and a $d \times d$ covariance matrix. The mixture weights α_k sum to one and are nonnegative.

As a flavor of K-means method, the objective function model-based clustering maximizes is the likelihood of $\mathbf{Y}$ given the model mixture parameters Θ, namely, $p(\mathbf{Y}|\Theta)$. In practice, the likelihood can be represented by any function of Θ that is proportional to the probability of the data $p(\mathbf{Y}|\Theta)$. In our application, the log of the likelihood of $\mathbf{Y}$ is applied,

$$\mathscr{L}(\Theta|\mathbf{Y}) = log\ p(\mathbf{Y}|\Theta) = \sum_i log \sum_k^K \alpha_k p_k(\mathbf{y}_i|\theta_k) \tag{8}$$

To this end, conducting model-based clustering is to learn the parameters of the K component models given the set of time series curves and then assign each curve to individual clusters. An EM algorithm [10] can be applied to learn the model parameters.

We use the B-spline regression mixture model in our work, and the model component takes a form as Equation (6). As a result, the regression model leads to a cluster-specific probabilistic density function for $\mathbf{y}_i$:

$$p_k(\mathbf{y_i}|\mathbf{s_i}, \theta_k) = \mathscr{N}(\mathbf{y_i}|\mathbf{M_i}\mathbf{p}_k,\ \sigma_k^2\mathbf{I}) \tag{9}$$

where $\mathbf{p}_k$ and $\sigma_k^2\mathbf{I}$ are the mean vector and covariance matrix of the kth Gaussian component model.Then, the EM algorithm is executed as follows.

- E-step We assume that z_i, associated with each $\mathbf{y}_i$, indicates the curve membership from one of the K clusters. In the E-step, the posterior $p(z_i|\mathbf{y}_i, \mathbf{s}_i)$ is calculated, which gives the probability that the i-th curve is generated by cluster z_i. The probability of $\mathbf{y}_i$ being generated by cluster k takes the form [15],

$$w_{ik} = p(z_i = k|\mathbf{y_i}, \mathbf{s_i}) \propto \alpha_k p_k(\mathbf{y_i}|\mathbf{s_i}) = \alpha_k \mathscr{N}(\mathbf{y_i}|\mathbf{M_i}\mathbf{p}_k,\ \sigma_k^2\mathbf{I}) \tag{10}$$

- M-step In the M-step, the likelihood (8) is maximized with respect to the parameters $\{\mathbf{p}_k,\ \sigma_k,\ \alpha_k\}$. The solutions [15] are given as

$$\mathbf{p}_k = \big[\sum_i w_{ik} M_i' M_i\big]^{-1} \sum_i w_{ik} M_i' \mathbf{y_i} \tag{11}$$

$$\sigma_k^2 = \frac{1}{\sum_i w_{ik}} \sum_i w_{ik} \|\mathbf{y_i} - M_i \mathbf{p}_k\|^2 \tag{12}$$

$$\alpha_k = \frac{1}{n} \sum_i w_{ik} \tag{13}$$

After obtaining these mixture model parameters, we can refer each time series curve to a cluster.

3.3 Interactive Re-clustering of Time Series Curves

How to initialize the number of mixture model components and their parameters is a critical problem in model-based clustering. In many real-world problems, the actual model size is unknown. Two families of model selection methods that help determine the cluster number are cross validation [29] and Bayesian model selection

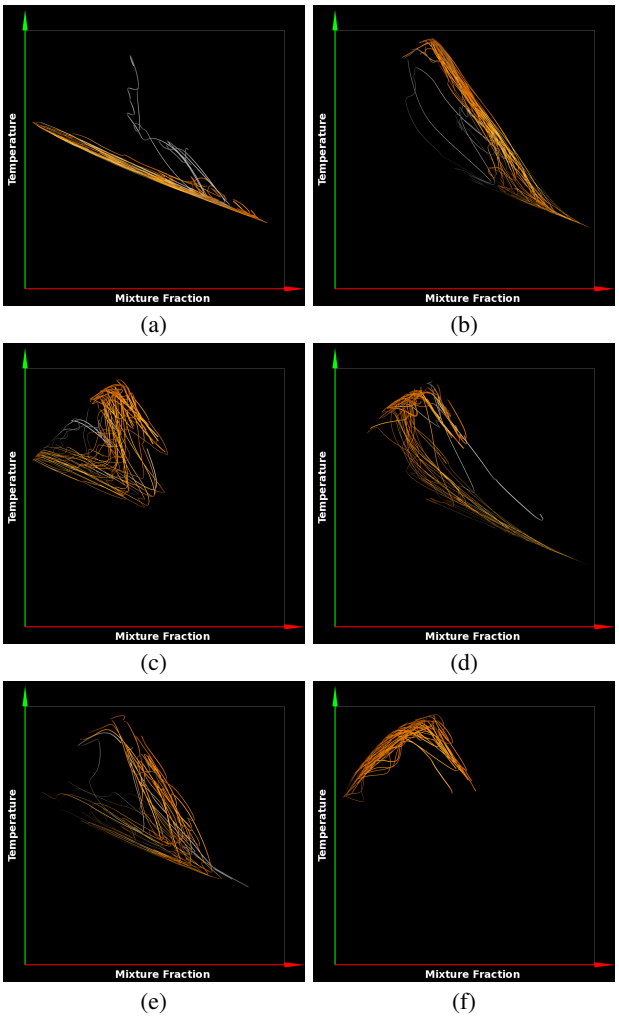


Figure 3: This figure shows six groups of curves, which are selected by the user as cluster prototypes for the re-clustering process. Comparing to those in Figure 2 (a), (d), (e), (h) and (i), the outlier curves (in gray) in (a), (b), (c), (d) and (e) are rejected using our mouse-based picking tool.

[3]. The initialization of mixture model parameters largely influences the quality of the clustering result. Since the standard EM algorithm adopts a strategy of randomly initializing the parameter values, it can only guarantee finding a local maximum, which may not correspond to satisfying clustering results. A number of variants of the EM algorithm have been studied to reduce its dependence on parameter initialization. These methods include the SEM algorithm [13] and the CEM algorithm [8]. Although these techniques reduce the effect of parameter initialization, they cannot solve the problem completely.

We solve the parameter initialization problem by introducing the domain experts' knowledge into the clustering process. In our application, we first empirically set the number of clusters as a comparatively large one in the hope of disclosing component patterns thoroughly. An extreme case is that if the cluster number is tuned the same as that of time series curves, all curve patterns are revealed. But we try to use the fewest number of clusters to represent patterns as completely as possible. Based on the preliminary results generated by automatic clustering, the user can browse through the clusters, and designate the number of clusters and what type of curves are in each one through interaction tools. We provide two mechanisms with which the user can select representative curves. The first is a mouse-based picking tool which selects the curve under the cursor during a user click. The second is a brushing tool, by using which the user sketches directly on the interface and all the underlying curves intersecting with the sketching are selected. Through users' repeating this selection, the number of clusters, together with the representative curves in each cluster, is determined. The component Gaussian model parameters, namely, the mean vector and covariance matrix, are learned from the prototype curves in each cluster. These parameters serve as the initial parameter values for the model-based clustering. In this way, the clustering process can be refined by integrating experts' domain knowledge.

3.4 Sub-clustering with Time Dimension

Until now, our clustering is focused on categorizing the general trends of time series curves in the 2D phase space. We partition the curves into clusters based on their geometric shapes and spatial positions. In that stage of cluster analysis, temporal relation among sample points on the curve is interpreted as the sequence order in the curve representation; nevertheless, how the curve change over time is not considered explicitly. It is possible that two curves possess similar correlation patterns in the 2D phase space, but are very different in how they vary over time. To handle this situation, we provide an option to further categorize time series curves in the clusters by considering the time dimension explicitly. Specifically, we extend the 2D phase space to 3D with time serving as a dimension, and the time series curve can be represented in the following way,

$$R = [(\mathbf{r}_1, t_1), (\mathbf{r}_2, t_2), ...(\mathbf{r}_n, t_n)] \tag{14}$$

where n, the number of sample timestamps in R, is defined as the length of R, and $\mathbf{r}_i$ is a data item of dimension d that is sampled at timestamp t_i. Comparing to the form (2), this representation treats time as an independent dimension.

The clustering method is similar to that introduced in Section 3.2, except that the time series curve is in 3D instead of 2D. By this means, we can further partition each cluster that is identified in the 2D phase space, and expose the patterns of temporal change in thermo-chemical state. Because a much clearer result is already generated after the 2D curve clustering step, we only use automatic clustering to partition the 3D time series curves.

4 Particle Data Analysis and Visualization in Dual-Space

4.1 Background Knowledge about Particle Data in Combustion Simulations

Regarding non-premixed combustion (fuel and oxidizer initially separated), in the phase space, the thermo-chemical state is largely a function of the *mixture fraction*. To a coarse approximation, analytic approaches based on using *mixture fraction* to form a coordinate system [27] are frequently employed to solve the governing equations of non-premixed flames. Even for such simplified approximations, the equations admit many solutions, two typical ones of which are the 'mixing' and 'burning' solutions. For the 'mixing' solution (without any flame present), temperature varies linearly with mixture fraction. The 'burning' solution takes a more complex representation. There is a maximum temperature around what is called the 'stoichiometric' *mixture fraction*, where the fuel and air are mixed in exactly the right proportions; at lower mixture fractions, there is extra air left over after the fuel is all gone, and at higher mixture fractions, there is extra fuel left over when the air is used up. Take the scatter plots of temperature - *mixture fraction* correlation in Figure 1 for an example, the points around the spot $P2$ are found in the fuel jet before it mixes with the oxidizer

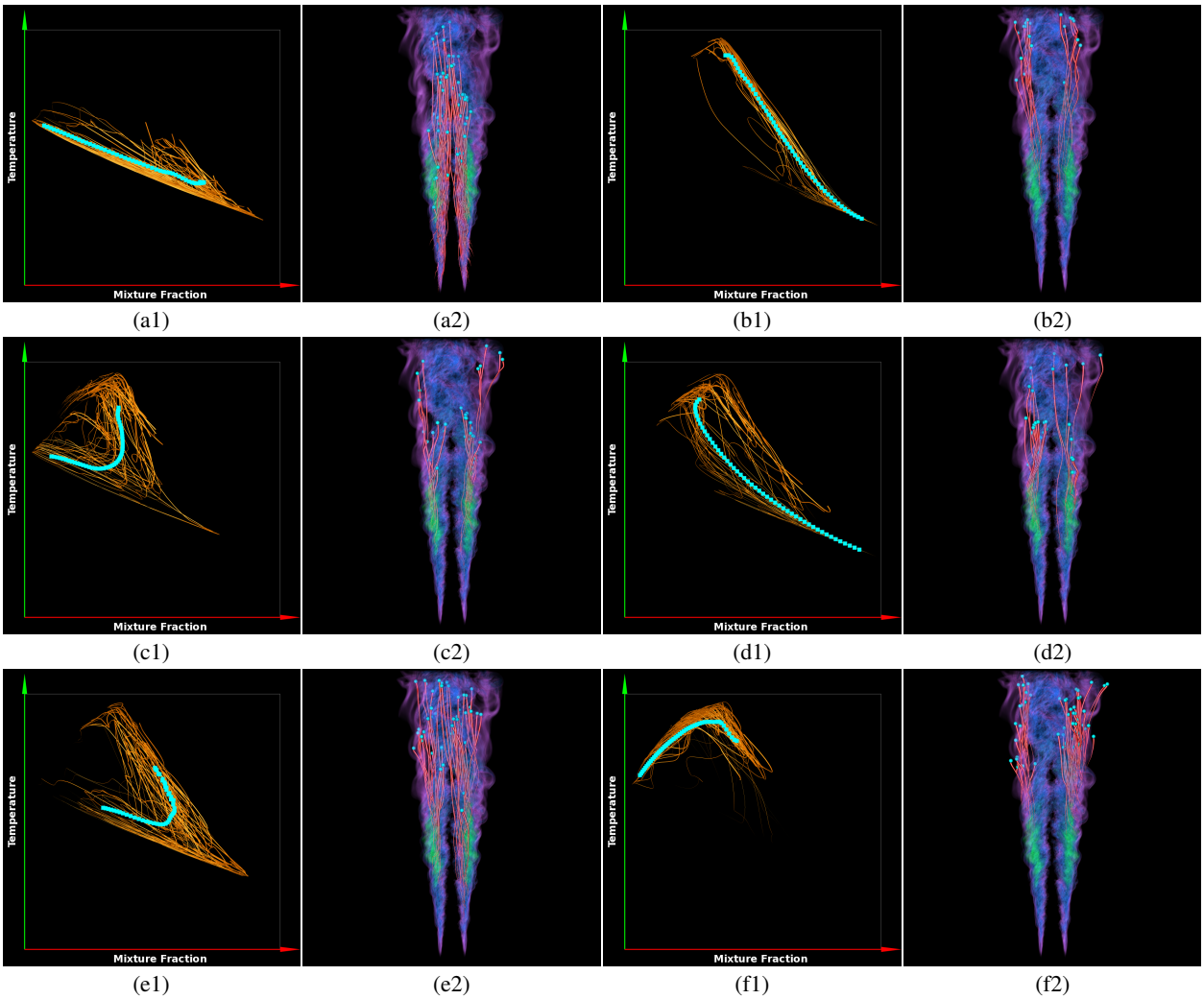


Figure 4: In this figure, (a1), (b1), (c1), (d1), (e1) and (f1) show the clustering results based on the user specified cluster prototypes; the light blue dots, calculated with the mean vector of component Gaussian models, represent the average trends of clusters. (a2), (b2), (c2), (d2), (e2) and (f2) show the particle trajectories in the physical space corresponding to (a1), (b1), (c1), (d1), (e1) and (f1) respectively. We can see that particles with distinct patterns of time series curves traverse differently in the physical space.

(the spot $P3$). Between these extreme spots $P2$ and $P3$, the majority of the points are found along one of two branches: a 'burning' branch and a 'mixing' branch. The green curve illustrates 'mixing' behavior - negatively correlated. The blue curve corresponds to the 'burning' solution: positive correlation for low *mixture fraction* and negative correlation for high *mixture fraction*; the red spot $P1$ is the stoichiometric *mixture fraction* point. The two correlation curves corresponding to the different branches of the temperature - *mixture fraction* solutions are relatively well understood; however, lots of particles are transitioning between the branches, which are less clear. Hence, combustion domain experts have a sound fundamental basis for expecting particle trajectories to move from the edges to the centre along either the mixing or burning branch, and to transition between the branches. In the following section, we use interactive clustering based on these expectations to either confirm (or deny) them, and to qualify the nature of the transition between the branches.

In one simulation, the Sandia DNS code S3D can generate several millions of particles with history records relating 3D particle positions and thermo-chemical states. To illustrate our method, we use a smaller data set of several hundreds of sampled time series curves. Figure 1 (a) shows an overview of the set of time series curves in the phase space, and Figure 1 (b) shows the corresponding trajectories in the physical space.

4.2 Automatic Clustering Results

In the 2D phase space, we first partition all time series curves into an estimated number of groups using the model-based clustering. Considering there are at least two distict groups of correlation cruves (the 'mixing' and 'burning' branches) and some others going between the two branches, we set the initial number of clusters to be ten, which is a comparatively large one, in the hope of disclosing component patterns thoroughly. In fact, to name an even larger number would not hurt the final results since we can revise this parameter in the following step. The aim of this automatic clustering is to address the visual clutter problem in a single visualization (Figure 1 (a)), and reveals interesting cluster patterns, as shown in Figure 2. Mostly, these initial clustering results may not be most satisfying. For example, Figure 2 (a) and (f) are of similar pattern and better to be combined together, as is the case for (b) and (j), (d)

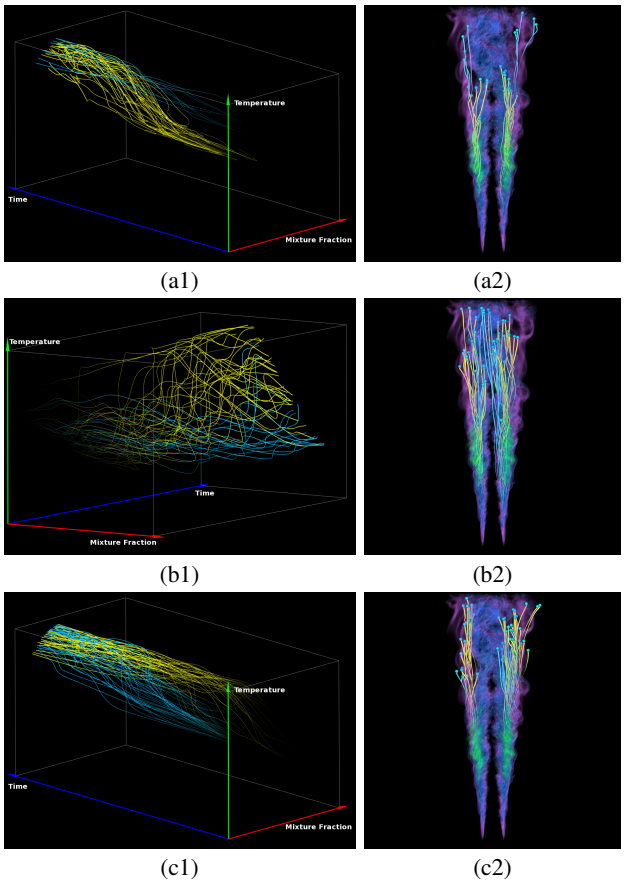


Figure 5: (a1), (b1) and (c1) show the temporal correlation curves corresponding to those in Figure 4 (c1), (e1) and (f1) respectively, and the 3D time series curves are clustered into two groups (rendered in yellow and blue); in (a2), (b2) and (c2), the particles with the yellow or blue trajectories belong to the clusters of the same color in (a1), (b1) and (c1)

and (g). In addition, clusters (a) and (f) contain obvious outliers.

4.3 Interactive Re-clustering Results

When analyzing the time series curves, domain experts usually have certain background knowledge and may suggest their preference to classify the data. Regarding our case, with the background knowledge introduced in 4.1, two clusters of time series curves we want to see include Figure 3 (a) and (f) which correspond to the 'mixing' and 'burning' branches. Aside from these cases, there is also a set of abnormal curves. Examples include Figure 3 (b) (c), (d) and (e). The domain experts are very interested in such groups of curves, which are not clearly understood yet. Consequently, we set these six groups of representative curves as cluster prototypes and re-cluster the whole data set. Figure 4 (a1), (b1), (c1), (d1), (e1) and (f1) show the re-clustered results in the phase space, and (a2), (b2), (c2), (d2), (e2), (f2) show the corresponding particle trajectories in the physical space. The trajectory groups demonstrate that particles with different patterns of time activity curves traverse distinctly in the physical space. After this interactive clustering, a much more reasonable and organized clustering result is generated.

4.4 Sub-clustering of Time Series Curves

Figure 5 (a1), (b1) and (c1) show the 3D time series curves, with the corresponding 2D projection in Figure 4 (c1), (e1) and (f1). We further partition each group of time series curves into two clusters, which are rendered with different colors. Figure 5 (a2), (b2) and (c2) show the particle trajectories in the physical space. By using the two-stage clustering in 2D and 3D sequentially, we categorize the patterns of temporal correlation between temperature and *mixture fraction*. Moreover, particles movements in the physical space are illustrated. The dual-space approach provides domain experts an effective way to relate how particle state changes in both phase and physical space.

4.5 Discussion

At this stage, the domain experts from the combustion field have been working closely with the visualization experts as the capabilities described in this paper are developed and deployed. In fact, there is an 'un-stimulated need' for this capability: the domain experts, two of whom are co-authors on this paper, were struggling to process the very data used to demonstrate the methodology in this paper. At the inception of this effort, the domain experts had several hypothesis about the nature of the particle trajectories as described in Section 4.1, but were unable to determine if the particle data was consistent with their expectations, nor were they able to present the particle data coherently to the combustion community. The ability of the clustering system to interactively partition the trajectories depending on suggested trajectories is key to hypothesis testing. With the current system, an expository movie has been made which the domain scientists have been actively using as an aid for discussing their simulation results with colleagues.

5 Conclusion and Future Work

In this paper, we present a dual-space approach to analyze particles' time series curves in the phase space and trajectories in the physical space for the combustion research. Regarding time series curves analysis, we use a two-stage clustering strategy to analyze the time series curves: in the first stage, time series curves are clustered in the 2D phase space to exhibit distinct bivariate correlations, which matches the combustion researchers' primary interests; in the second stage, within each cluster, we treat the temporal bivariate correlations as 3D curves (temperature, *mixture fraction* and time serve as the three dimensions) and partition them in several groups to further reveal how the correlation changes over time. We also study how to integrate the users' domain knowledge into the model-based clustering process. This integration improves the clustering results by using domain knowledge to initialize the algorithm parameters.

Our dual-space technique highlights the relationship between the particle trajectories in phase and physical space, which provides combustion scientists with detailed information regarding the evolution of fluid parcels traversing a turbulent autoignitive environment. Although we have designed our approach that targets to facilitate combustion studies, we expect that the basic methodology can be applied to other scientific simulations involving particle trajectory analysis work.

Interaction is the key in data analysis tasks, not only for manipulating visual results, but also for steering automatic clustering. In the future, we plan to improve our current design of interaction tools and develop new ones to enrich users' choices in expressing their domain knowledge to steer clustering. For instance, free sketching is a promising interactive method, by which users can specify different curve patterns according to their knowledge. These curve patters can then be used to guide the clustering algorithm. Currently, we utilize model-based clustering to analyze bivariate time series data. This algorithm can be easily extended to handle multivariate time series data clustering. But how to visualize and interactively manipulate the clustering results of multivariate time series curves needs further study. Thus, we will place efforts on developing proper visualization and interaction methods to represent and operate the high-dimensional clustering results. Moreover, we will

also address the large data issue with parallel computing so that our method can adopt to the ever-increasing large simulation data.

Acknowledgements

This research was supported in part by the U.S. National Science Foundation through grants ACI-0749227, OCI-0950008, and OCI-0850566, and the U.S. Department of Energy through the SciDAC program with Award No. DE-FC02-06ER25777.

The authors wish to thank all the reviews for their valuable comments and suggestions.

References

[1] W. Aigner, S. Miksch, W. Müller, H. Schumann, and C. Tominski. Visual methods for analyzing time-oriented data. *IEEE Transactions on Visualization and Computer Graphics*, 14(1):47–60, 2008.

[2] H. Akiba and K.-L. Ma. A tri-space visualization interface for analyzing time-varying multivariate volume data. In *EuroVis*, pages 115–122, 2007.

[3] J. D. Banfield and A. E. Raftery. Model-based Gaussian and non-Gaussian clustering. *Biometrics*, 49(3):803–821, 1993.

[4] S. Basu, I. Davidson, and K. Wagstaff. *Constrained Clustering: Advances in Algorithms, Theory, and Applications*. Chapman & Hall/CRC, 2008.

[5] G. K. Batchelor. *An introduction to fluid dynamics*. Cambridge University Press, Cambridge, 1967.

[6] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *KDD Workshop*, pages 359–370, 1994.

[7] E. Bertini and D. Lalanne. Investigating and reflecting on the integration of automatic data analysis and visualization in knowledge discovery. *SIGKDD Explorations*, 11(2):9–18, December 2009.

[8] G. Celeux and G. Govaert. A classification em algorithm for clustering and two stochastic versions. *Comput. Stat. Data Anal.*, 14(3):315–332, 1992.

[9] J. H. Chen, A. Choudhary, B. de Supinski, M. DeVries, E. R. Hawkes, S. Klasky, W. K. Liao, K. L. Ma, J. Mellor-Crummey, N. Podhorszki, R. Sankaran, S. Shende, and C. S. Yoo. Terascale direct numerical simulations of turbulent combustion using s3d. *Comput. Sci. Disc.*, 2:015001, 2009.

[10] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1):1–38, 1977.

[11] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh. Querying and mining of time series data: experimental comparison of representations and distance measures. *Proc. VLDB Endow.*, 1(2):1542–1552, 2008.

[12] Z. Fang, T. Möller, G. Hamarneh, and A. Celler. Visualization and exploration of time-varying medical image data sets. In *GI '07: Proceedings of Graphics Interface 2007*, pages 281–288, New York, NY, USA, 2007. ACM.

[13] J. D. G. Celeux. The sem algorithm: a probabilistic teacher algorithm derived from the em algorithm for the mixture problem. *Comput. Statist. Quar. 2*, pages 73–82, 1985.

[14] S. Gaffney and P. Smyth. Trajectory clustering with mixtures of regression models. In *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 63–72, New York, NY, USA, 1999.

[15] S. Gaffney and P. Smyth. Joint probabilistic curve clustering and alignment. In *NIPS*, 2004.

[16] S. J. Gaffney, A. W. Robertson, P. Smyth, S. J. Camargo, and M. Ghil. Probabilistic clustering of extratropical cyclones using regression mixture models. Technical report, Climate Dynamics, 2006.

[17] J. Han. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[18] J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. *JSTOR: Applied Statistics*, 28(1):100–108, 1979.

[19] H. Hochheiser and B. Shneiderman. Interactive exploration of time series data. In *Discovery Science*, pages 441–446, 2001.

[20] D. A. Keim, F. Mansmann, and J. Thomas. Visual analytics: How much visualization and how much analytics? *SIGKDD Explorations*, 11(2):5–8, December 2009.

[21] E. Keogh. A decade of progress in indexing and mining large time series databases. In *VLDB '06: Proceedings of the 32nd international conference on Very large data bases*, pages 1268–1268. VLDB Endowment, 2006.

[22] Z. Konyha, K. Matkovic, D. Gracanin, M. Jelovic, and H. Hauser. Interactive visual analysis of families of function graphs. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1373–1385, 2006.

[23] S. H. Lamb. *Hydrodynamics (6th ed.)*. Cambridge University Press, Cambridge, 1994.

[24] 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.

[25] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[26] P. Muigg, J. Kehrer, S. Oeltze, H. Piringer, H. Doleisch, B. Preim, and H. Hauser. A four-level focus+context approach to interactive visual analysis of temporal features in large scientific data. *Comput. Graph. Forum*, 27(3):775–782, 2008.

[27] N. Peters. Laminar diffusion flamelet models in non-premixed turbulent combustion. *Progress in Energy and Combustion Science*, 10:319–339, 1984.

[28] T. Schreck, J. Bernard, T. Von Landesberger, and J. Kohlhammer. Visual cluster analysis of trajectory data with interactive kohonen maps. *Information Visualization*, 8(1):14–29, 2009.

[29] P. Smyth. Model selection for probabilistic clustering using cross-validated likelihood. *Statistics and Computing*, 10(1):63–72, 2000.

[30] J. J. Van Wijk and E. R. Van Selow. Cluster and calendar based visualization of time series data. In *INFOVIS '99: Proceedings of the 1999 IEEE Symposium on Information Visualization*, page 4, Washington, DC, USA, 1999. IEEE Computer Society.

[31] T. Warren Liao. Clustering of time series data-a survey. *Pattern Recogn.*, 38(11):1857–1874, 2005.

[32] J. Woodring and H.-W. Shen. Semi-automatic time-series transfer functions via temporal clustering and sequencing. *Comput. Graph. Forum*, 28(3):791–798, 2009.

[33] Y. Xiong and D.-Y. Yeung. Time series clustering with arma mixtures. *Pattern Recognition*, 37(8):1675–1689, 2004.

[34] P. Yeung. Lagrangian investigations of turbulence. *Annual Review Of Fluid Mechanics*, 34:115–142, 2002.