

Visual Analysis of Particle Behaviors to Understand Combustion Simulations

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*

A dual-space method enables effective visual analysis of particles' spatial movement and attribute evolution. Intuitive interaction tools integrate users' domain knowledge to steer classification. This method has been used to analyze combustion simulations and is applicable to other scientific simulations involving particle-data analysis.

Advanced combustion research is essential to designing more efficient engines. Next-generation engines will operate in nonconventional, mixed-mode, and turbulent conditions. Combustion processes in such an environment, combined with new physical and chemical fuel properties, feature complicated interactions that are poorly understood at a fundamental level. Recently, Sandia National Laboratories scientists have instrumented their simulations with particles to capture and better understand the turbulent dynamics in combustion processes. So, how to analyze and visualize these particles' temporal behaviors from different aspects is critical to understanding combustion.

When visualizing a large number of moving particles, we confront two main issues. The first is what properties of the particle data to visualize; the other is how to deal with the large data. To conduct a comprehensive study of particle behaviors, a visualization system must be able to present the temporal

variation of particle properties. Conventionally, the 3D simulation domain, in which particles are advected, is called the *physical space*; the attribute domain, in which particle attributes evolve, is called the *phase space*. A particle's spatial movement in physical space is called its *trajectory*, and its attribute variation in phase space is called the *attribute evolution curve*. No matter whether we're dealing with the physical or phase space, we can record a particle's history as a sequence of points.

To analyze and visualize particles' spatial movement and attribute evolution, we've developed a *dual-space* method. "Dual space" means a combination of the physical and phase spaces, in which we depict both trajectories and attribute evolution curves as lines. Visualizing a large bulk of such lines would be a mess, considering the data's sheer size and a constrained display space. Dense lines intertwine and interesting features become hidden, which prohibit scientists from testing hypotheses or discovering phenomena.

To deal with large data, our dual-space system incorporates semisupervised learning.¹ With domain experts' guidance, the system classifies the set of lines into distinct groups. Each group has a different pattern and can be visualized in a clutter-free manner, with its features clearly vis-

Time-Varying Data Classification and Visualization

Visual analysis of time-varying data has been broadly studied by visualization researchers. Several researchers have developed automatic approaches. For example, in medical-data analysis, Zhe Fang and his colleagues considered time-varying data as a 3D array in which each voxel contains a time-activity curve.¹ They defined three similarity metrics to cluster and visualize these curves. Jarke van Wijk and Edward van Selow proposed a cluster-and-calendar-based analytical system to explore and visualize univariate time series data.² They used hierarchical clustering to categorize daily time series patterns and visualized them on a yearly calendar.

Researchers have also developed interactive approaches to extract interesting patterns from time-varying data. TimeSearcher let users retrieve time series by creating queries.³ Retrieval employed TimeBoxes, rectangular query locators that specified interesting time series. Hiroshi Akiba and Kwan-Liu Ma introduced a tri-space visualization interface for examining multivariate time-varying data.⁴ Zoltán Konyha and his colleagues' system for exploring and analyzing function graphs combined established visualization techniques, linked views, and advanced brushing features.⁵ To deal with overdrawing and visual clutter when depicting large numbers of function graphs, Philipp Muigg and his colleagues developed a four-level "focus+context" interactive visualization technique, with the context information for orientation and three levels of focus in every attribute view.⁶

In practice, automatic and interactive approaches both have advantages and limitations. There has been a trend toward integrating them for analysis and visualization of large, complex datasets.^{7,8} Jonathan Woodring and Han-Wei Shen proposed a technique to semiautomatically generate transfer functions for time-varying data through temporal clustering and sequencing.⁹ Teng-Yok Lee and Shen presented an algorithm that identifies important trends in relationships among the variables on the basis of how the variables' values change over time and how those changes are related to each other in different spatial regions and time intervals.¹⁰ Tobias Schreck and his colleagues proposed a self-organizing-map clustering algorithm that lets users visually control and monitor the computation to leverage their domain knowledge.¹¹

Our research (see the main article) also integrates automatic data analysis with user interaction that exploits human domain knowledge. The automatic technique—clustering of time series curves—has been a research focus in both the visualization and data-mining communities, and researchers have made great advances. To gain a better understanding, see the comprehensive surveys on visual analysis of time-oriented data¹² and time series data mining.^{13,14}

References

1. Z. Fang et al., "Visualization and Exploration of Time-Varying Medical Image Data Sets," *Proc. Graphics Interface 2007*, ACM Press, 2007, pp. 281–288.
2. J.J. van Wijk and E.R. van Selow, "Cluster and Calendar Based Visualization of Time Series Data," *Proc. 1999 IEEE Symp. Information Visualization (InfoVis 99)*, IEEE CS Press, 1999, pp. 4–9.
3. H. Hochheiser and B. Shneiderman, "Interactive Exploration of Time Series Data," *Proc. 4th Int. Conf. Discovery Science*, Springer, 2001, pp. 441–446.
4. H. Akiba and K.-L. Ma, "A Tri-space Visualization Interface for Analyzing Time-Varying Multivariate Volume Data," *Proc. Eurographics/IEEE VGTC Symp. Visualization*, Eurographics Assoc., 2007, pp. 115–122.
5. Z. Konyha et al., "Interactive Visual Analysis of Families of Function Graphs," *IEEE Trans. Visualization and Computer Graphics*, vol. 12, no. 6, 2006, pp. 1373–1385.
6. P. Muigg et al., "A Four-Level Focus+Context Approach to Interactive Visual Analysis of Temporal Features in Large Scientific Data," *Computer Graphics Forum*, vol. 27, no. 3, 2008, pp. 775–782.
7. E. Bertini and D. Lalanne, "Investigating and Reflecting on the Integration of Automatic Data Analysis and Visualization in Knowledge Discovery," *SIGKDD Explorations*, vol. 11, no. 2, 2009, pp. 9–18.
8. D.A. Keim, F. Mansmann, and J. Thomas, "Visual Analytics: How Much Visualization and How Much Analytics?" *SIGKDD Explorations*, vol. 11, no. 2, 2009, pp. 5–8.
9. J. Woodring and H.-W. Shen, "Semi-automatic Time-Series Transfer Functions via Temporal Clustering and Sequencing," *Computer Graphics Forum*, vol. 28, no. 3, 2009, pp. 791–798.
10. T.-Y. Lee and H.-W. Shen, "Visualization and Exploration of Temporal Trend Relationships in Multivariate Time-Varying Data," *IEEE Trans. Visualization and Computer Graphics*, vol. 15, no. 6, 2009, pp. 1359–1366.
11. T. Schreck et al., "Visual Cluster Analysis of Trajectory Data with Interactive Kohonen Maps," *Information Visualization*, vol. 8, no. 1, 2009, pp. 14–29.
12. W. Aigner et al., "Visual Methods for Analyzing Time-Oriented Data," *IEEE Trans. Visualization and Computer Graphics*, vol. 14, no. 1, 2008, pp. 47–60.
13. T.W. Liao, "Clustering of Time Series Data—a Survey," *Pattern Recognition*, vol. 38, no. 11, 2005, pp. 1857–1874.
14. E. Keogh, "A Decade of Progress in Indexing and Mining Large Time Series Databases," *Proc. 32nd Int'l Conf. Very Large Data Bases (VLDB 06)*, VLDB Endowment, 2006, pp. 1268–1268.

ible. (For more information on visualizing time-varying data, see the related sidebar.)

While developing our method, we worked closely with combustion scientists (two of whom

are coauthors of this article) to create a prototype system.² This system successfully characterized different particle behaviors in the dual space. So, the combustion scientists wanted to enhance this

Regression Mixture Models

Regression models can be used for curve-like time series data modeling. Model-based clustering often applies a mixture of regression models.

Linear Regression

Regression fits a curve through a set of points using some goodness-of-fit criterion. One of the most common types of regression is linear regression. Let x be an independent variable and $\mathbf{p}(x)$ be an unknown function of x that we want to approximate. Assume there are R observations, and each has D dimensions; that is, the values of $\mathbf{p}(x)$ measured at the specified values of x_r are

$$\mathbf{p}(x_r) = \mathbf{p}_r, r = 1, \dots, R.$$

Let a sequence of points $(\mathbf{p}(x_1), \mathbf{p}(x_2), \dots, \mathbf{p}(x_R))$ represent a curve. Regarding each dimension of this curve, the idea behind linear regression is to model $\mathbf{p}(x)$ by a linear combination of Q basis functions:

$$\mathbf{p}(x) \approx \beta_1\psi_1(x) + \dots + \beta_Q\psi_Q(x),$$

where β represents a $1 \times D$ vector and ψ represents a scalar function of x .

For polynomial basis functions,

$$\mathbf{p}(x) \approx \beta_0 + \beta_1x + \dots + \beta_Qx^Q. \tag{A}$$

We can write Equation A as

$$l = \mathbf{X}\beta + \varepsilon, \tag{B}$$

where l is an $R \times D$ matrix representing a line of length R in D -dimensional space, β is a $Q \times D$ matrix of regression coefficients, and ε is an $R \times D$ noise matrix. \mathbf{X} is the usual $R \times Q$ Vandermonde regression matrix:

$$\mathbf{X} = \begin{bmatrix} x_1^0 & x_1^1 & x_1^2 & \dots & x_1^Q \\ x_2^0 & x_2^1 & x_2^2 & \dots & x_2^Q \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_R^0 & x_R^1 & x_R^2 & \dots & x_R^Q \end{bmatrix}.$$

Model Parameter Recovery

Given a set of lines originating from K classes, each of which is described by a linear regression associated with a Gaussian error term as shown in Equation B, we get a mixture of K component regression models. Let p_k denote the probability at which the component model θ_k generates a line; the mixture density for generating line l is

$$p(l|\Theta) = \sum_k \alpha_k p_k(l|\theta_k),$$

where α_k denotes the probability of class k , which is non-negative, and all component probabilities (for $k = 1 \dots K$) sum to one. θ_k indicates the parameters of component model k . Each component θ_k contains regression coefficients β_k and the Gaussian covariance parameter δ_k . So, the mixture model is $\Theta = \{\theta_1, \dots, \theta_k\}$. $p_k(l|\theta_k)$ is the probability of compo-

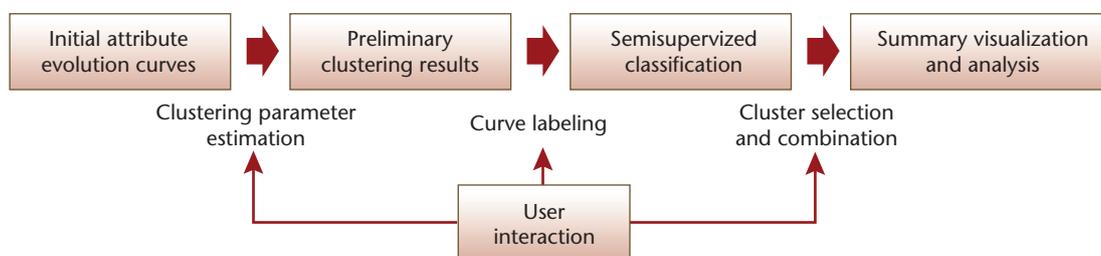


Figure 1. The major steps of our method for analyzing and visualizing particle data in the dual space. This process iterates until it achieves a satisfactory categorization.

system and incorporate it into their overall scientific validation and discovery process. This article describes both the dual-space system's design and the enhancements we've made. In particular, we replaced heuristic interactive clustering² with semi-supervised learning because the latter is more flexible in cluster model updating and generates results that conform more closely to user specifications.

Visual Analysis in the Dual Space

Figure 1 shows our method's main steps. To cat-

egorize attribute evolution curves, we follow a *cluster-label-classify* strategy. First, automatic clustering discloses the patterns hidden in the original line data and provides an overview of line features. Next, users manipulate groups of lines and launch a semisupervised learning engine to refine the classification of the curves. This process iterates until it achieves a satisfactory categorization. Finally, for the different particle groups, the system displays and links the phase-space and physical-space views so that users can examine their connections.

nent model k generating l . In our research (see the main article), the component model takes the form of Equation B. As a result, the regression model leads to a class-specific probabilistic density function for l :

$$p_k(l|\theta_k) = \mathcal{N}(l|\mathbf{X}\beta_k, \sigma_k^2\mathbf{I}) \\ = \prod_{r=1}^R \mathcal{N}(p_r|\mathbf{X}_r\beta_k, \sigma_k^2\mathbf{I}),$$

where $\mathcal{N}(\cdot)$ is a D -dimensional Gaussian probability density function; $\mathbf{X}\beta_k$ and $\sigma_k^2\mathbf{I}$ are the mean vector and covariance matrix of the k th Gaussian density function.

To find the model parameters Θ under the criterion of the maximum-likelihood estimate, the expectation maximization (EM) algorithm provides a locally optimal solution. In practice, given a line dataset $\mathbf{L}(\mathbf{L} = \{l_1, l_2, \dots, l_N\})$, we can represent the likelihood $\mathcal{L}(\Theta|\mathbf{L})$ by any function of Θ that's proportional to the probability $p(\mathbf{L}|\Theta)$. In our application, we apply the log of likelihood \mathcal{L} :

$$\mathcal{L}(\Theta|\mathbf{L}) = \log p(\mathbf{L}|\Theta) = \sum_n \log \sum_k \alpha_k p_k(l_n|\theta_k).$$

The EM algorithm has two stages. The first initializes the probability p_{ik} of each line l_i belonging to class k as a random number, constrained by

$$\sum_{k=1}^K p_{ik} = 1.$$

Mixture Models

A mixture model is a powerful, flexible probabilistic model in machine learning. It assumes that the dataset derives from a mixture of K component models corresponding to K classes. Each component model has an associated probabilistic density function, such as a Gaussian or multinomial distribution. Once we obtain those distributions' parameters, we know how to do clustering or classification. In other words, clustering or classification with mixture models is about how to recover these model parameters.

Researchers have studied several optimization methods in depth to find locally optimal model parameters. A particularly important method is expectation maximization (EM).³ We use polynomial-regression mixture models to cluster and classify particle line data with the EM algorithm. (For more on this, see the "Regression Mixture Models" sidebar.)

Automatic Clustering

Using mixture models for clustering is often called

The second stage iteratively performs two steps to find the locally optimal solution. The E-step calculates the expectation of the log of likelihood \mathcal{L} ,

$$Q = E[\mathcal{L}(\Theta|\mathbf{L})] = \sum_n \sum_k w_{nk} \log \alpha_k \mathcal{N}(l_n|\mathbf{X}\beta_k, \sigma_k^2\mathbf{I}).$$

The posterior w_{nk} that gives the probability of the n th line being generated by class k is¹

$$w_{nk} = p(k|l_n) \propto \alpha_k^{old} p_k(l_n) = \alpha_k^{old} \mathcal{N}(l_n|\mathbf{X}\beta_k^{old}, \sigma_k^{old^2}\mathbf{I}),$$

where β_k^{old} , $\sigma_k^{old^2}$, and α_k^{old} are model parameters calculated in the last iteration. The M-step maximizes the expectation of the log of likelihood \mathbf{L} with respect to the model parameters $\{\beta_k, \sigma_k^2, \alpha_k\}$. The solutions are¹

$$\beta_k = \left[\sum_n w_{nk} \mathbf{X}'_n \mathbf{X}_n \right]^{-1} \sum_n w_{nk} \mathbf{X}'_n l_n, \\ \sigma_k^2 = \frac{1}{\sum_n w_{nk}} \sum_n w_{nk} \|l_n - \mathbf{X}_n \beta_k\|^2, \\ \alpha_k = \frac{1}{N} \sum_n w_{nk}.$$

Reference

1. S. Gaffney and P. Smyth, "Joint Probabilistic Curve Clustering and Alignment," *Advances in Neural Information Processing Systems 17*, MIT Press, 2004, pp. 473–480.

model-based clustering. This clustering method benefits our analysis tasks in two ways:

- It's computationally efficient and possesses a complexity of order n , where n is the number of items to cluster.
- It provides a principled way to cluster lines with different lengths, which is a favorable characteristic in line data clustering.

Model-based clustering is basically a generalization of the K -means algorithm. Polynomial-regression model-based clustering assumes that the whole set of lines derives from a mixture model of K polynomial-regression components corresponding to K clusters. Each component model has an associated Gaussian error term. The EM algorithm solves the model-based clustering problem by recovering parameters of regression mixture models.

Users must specify the number of clusters into which the data are partitioned. Because the goal isn't to create clustering results accurately but to reveal line patterns comprehensively during automatic

```

Input: A dataset of labeled and unlabeled lines,  $\mathbf{L}$ 
Output:  $K$  classes of lines
1: // Preprocessing
2: Smooth each line with the uniform B-spline model
3: Sample each line to obtain its vector descriptor
4: // Initialization
5: Calculate model parameters  $\Theta$  using the labeled lines
6: while true do
7:   // E-Step
8:   Determine the probability of each labeled or unlabeled line belonging to a class  $k$ 
9:   if likelihood  $\mathcal{L}(\Theta|\mathbf{L})$  converges then
10:    break
11:   end if
12:   // M-Step
13:   Calculate  $\Theta$  using both the labeled and unlabeled lines
14: end while
15: // Generating final classification
16: Put each line in the class  $k$  with the highest probability

```

Figure 2. Semisupervised classification with regression mixture models. This process employs the labeled lines to initialize the number of mixture model components and their parameters. It uses both labeled and unlabeled lines during iterated expectation maximization to classify all data, integrating users' knowledge to improve the preliminary clustering results.

clustering, users only need to empirically set the cluster number. A comparatively larger number tends to produce more line patterns. An extreme case is that if the cluster number is the same as the number of attribute evolution curves, the system will display all the curve patterns. Users should select the possible lowest number of clusters that represents patterns as completely as possible, so as to ease later interaction.

Semisupervised Classification

Machine-learning algorithms fall into roughly three categories based on how much human supervision they involve:

- unsupervised learning (for example, clustering and outlier detection), in which all data are unlabeled;
- supervised learning (for example, classification and regression), in which all data are labeled; and
- semisupervised learning, which comes somewhere between the other two.

Unsupervised learning needs no human supervision and extracts hidden structures directly from the unlabeled data. However, it doesn't guarantee results that meet user requirements.

Acquiring labeled data often requires experienced agents' knowledge and is usually time-consuming. When complex large data are involved, as in our case, labeling's cost makes supervised learning prohibitive.

On the other hand, semisupervised learning exploits the strengths of both supervised and unsupervised learning. It can classify data using a limited amount of labeled data and a larger amount of unlabeled data. This combination produces better learning behaviors in that the labeled data incor-

porate experts' domain knowledge and the unlabeled data help identify data patterns.

We adopt semisupervised learning to classify particle attribute evolution curves to refine the automatic-clustering results. Users browse through the preliminary clusters and designate the number of groups and the type of curves in each group. Our system provides two tools for selecting representative curves. With the mouse-based picking tool, users click on a curve to select it. With the brushing tool, users sketch directly on the interface, and the tool selects the underlying curves that intersect with the sketching. As users repeat this selection, the system determines the number of clusters and the representative curves in each cluster. In other words, user selection implicitly labels representative lines as distinct groups according to the users' domain knowledge.

Figure 2 illustrates our semisupervised-classification algorithm. First, it smoothes the lines with the B-spline model and resamples them to obtain their vector descriptors. Next, it calculates each cluster's initial model parameters on the basis of the representative lines the user labeled. Then, the algorithm performs iterative EM to find each cluster's locally optimal model parameters.

The semisupervised classification exploits the labeling information in two ways. First, it employs the labeled lines to initialize the number of mixture model components and their parameters, which is critical in recovering mixture models. Second, it uses both labeled and unlabeled lines in the EM iteration to classify all data, integrating users' knowledge to improve the preliminary clustering results.

Data Visualization and Analysis

Visualizing particles' temporal behaviors in the dual

Direct Numerical Simulation for Combustion Research

Scientists use direct numerical simulation (DNS) to capture and describe the key turbulence chemistry interactions. Sandia National Laboratories developed S3D, a massively parallel solver, to solve the DNS governing equations originating from a Eulerian viewpoint.¹ The Eulerian specification of the flow field focuses on specific locations in the space through which the fluid flows over time.^{2,3}

The Lagrangian viewpoint is another common method to describe fluid flows. In the Lagrangian specification, the observer follows a fluid parcel as it moves through space and time.^{2,3} 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

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

where $\vec{u}(\vec{x}_0, t)$ is the instantaneous field velocity at $\vec{x}(\vec{x}_0, t)$.⁴

The transport of combustion turbulence is dominated by advective transport, so the Lagrangian description is natural and useful for the treatment of turbulent mixing.⁴ At Sandia National Laboratories, recent combustion simulations of a turbulent lifted autoignitive ethylene-air jet flame in a hot-air coflow have employed particles originating from both the fuel and oxidizer sources.¹

These simulations provided a Lagrangian description of the combustion environment. The passive tracer particles were 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 determined the particle velocity from the Eulerian solution. When the simulations integrated the particle position, they also saved the thermochemical state (temperature, composition, and so on), interpolated from the Eulerian grid to the particle positions. In this way, DNS provides a set of particles, each of which contains a record of the history of its movement positions and changing thermochemical states. The main article briefly introduces these simulations.

References

1. J.H. Chen et al., "Terascale Direct Numerical Simulations of Turbulent Combustion Using S3D," *Computational Science and Discovery*, vol. 2, no. 015001, 2009.
2. G.K. Batchelor, *An Introduction to Fluid Dynamics*, Cambridge Univ. Press, 1967.
3. S.H. Lamb, *Hydrodynamics*, 6th ed., Cambridge Univ. Press, 1994.
4. P. Yeung, "Lagrangian Investigations of Turbulence," *Ann. Rev. Fluid Mechanics*, vol. 34, 2002, pp. 115–142.

space helps scientists explore, understand, and present their data. Direct line rendering provides an intuitive visual expression of how the particles traverse the physical space and how particles' attributes interact in the phase space. With the classification results, scientists can check different line patterns and use them to test hypotheses or discover phenomena.

In the phase space, on the basis of the classification of attribute evolution curves, we directly visualize each class of lines to demonstrate the distinct attribute variation patterns. Because each class has an associated mean regression curve, we provide an abstract visualization that shows each class's line trend. By examining each class, scientists can perceive a clear overview of the line patterns.

In the physical space, we incorporate line rendering and volume rendering to visualize the trajectories with respect to each particle cluster. The system embeds trajectories into the surrounding instantaneous field data at each time step and can animate the particle movement simultaneously in the phase and physical spaces.

The two views in the dual-space visualization appear in separate windows in the same application. By being able to see the data in two different views simultaneously, users gain a better understanding

of the connections between particle movement and attribute evolution. In this way, they can examine each particle category's movement pattern individually to reveal the particles' evolution and the correlation between the particle attributes.

Two Case Studies

For non-premixed combustion (that is, the fuel and oxidizer are initially separate), the attribute evolution state is largely a function of the *mixture fraction*. To a coarse approximation, to solve the governing equations of non-premixed flames, scientists frequently employ analytic approaches that use the mixture fraction to form a coordinate system.⁴

We evaluated our dual-space system's effectiveness by analyzing how the temperature variable and hydroxide (OH) species interact with the mixture fraction. In one simulation, Sandia's direct numerical simulation (DNS) code S3D can generate several millions of particles with history records relating 3D particle positions and attribute evolution states. To illustrate our method, we used a smaller dataset of several hundreds of particle records. (For details on DNS and S3D, see the "Direct Numerical Simulation for Combustion Research" sidebar.)

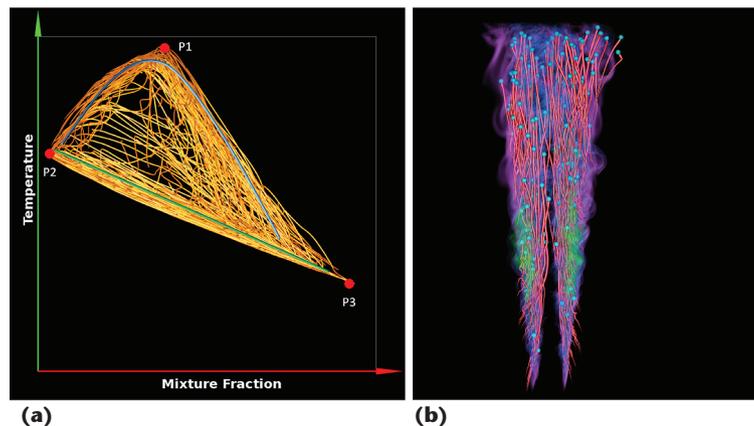


Figure 3. The relation between temperature and the mixture fraction, which are two key parameters in the combustion simulation of a turbulent lifted autoignitive ethylene/air jet flame. (a) The attribute evolution curves in the phase space. (b) The corresponding particle trajectories in the physical space, with volume rendering of the hydroperoxy field. Both images show much clutter; users will have difficulty perceiving detailed correlation patterns.

Temperature and the Mixture Fraction

Two typical temporal relations of temperature and the mixture fraction are the mixing and burning solutions. For the mixing solution (in which no flame is present), the temperature varies linearly with the mixture fraction.

The burning solution takes a more complex representation. The maximum temperature occurs around the *stoichiometric* mixture fraction, in which the fuel and air are mixed in exactly the right proportions. At lower mixture fractions, air remains after the fuel is gone; at higher mixture fractions, fuel remains when the air is gone.

Figure 3a shows an overview of the attribute evolution curves in the phase space; Figure 3b shows the corresponding trajectories in the physical space. In Figure 3a, the points around spot P2 are found in the fuel jet before it mixes with the oxidizer at P3. Between these extreme spots, most of the points occur along one of two branches: a mixing branch and a burning branch. The green curve illustrates mixing behavior, which has a negative correlation between temperature and mixture fraction. The blue curve illustrates burning behavior, which has a positive correlation for a low mixture fraction and a negative correlation for a high mixture fraction. P1 is the stoichiometric mixture fraction point.

The two correlation curves corresponding to the different branches of the temperature and mixture fraction solutions are relatively well understood. However, many particles are transitioning between the branches and are less clear. So, combustion experts have a sound fundamental basis for expecting particle trajectories to move from the edges to the center along either the mixing or burning branch and to transition between the

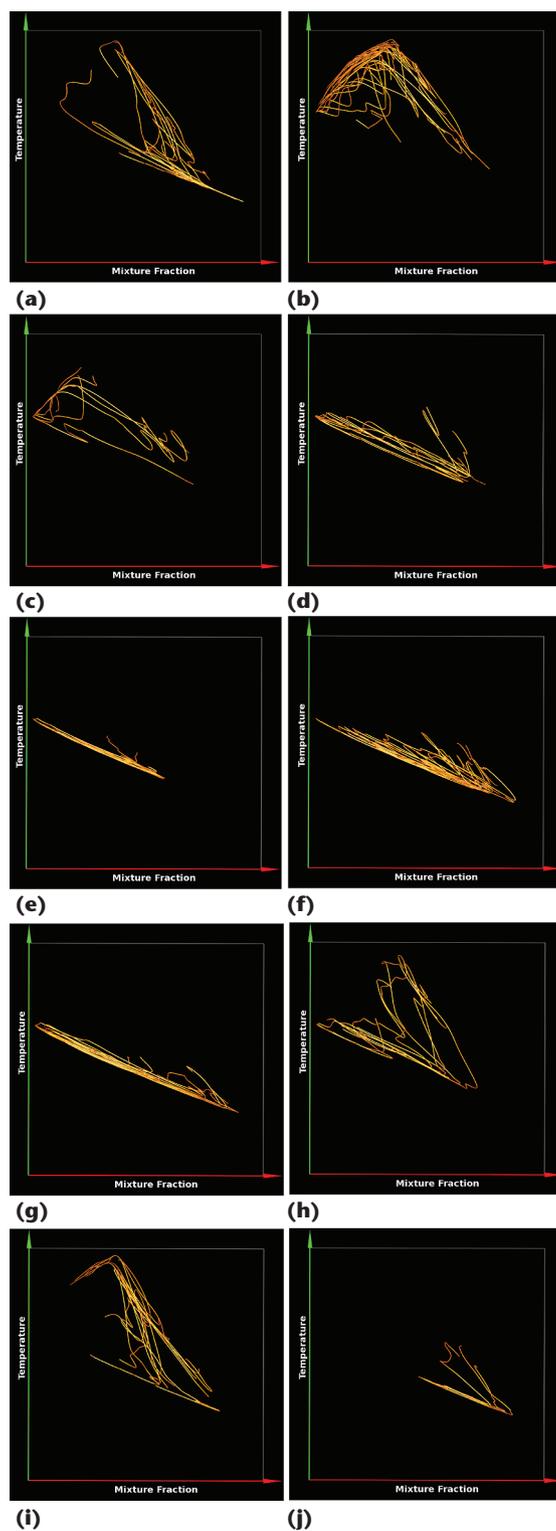


Figure 4. Automatic model-based clustering generated these 10 groups of attribute evolution curves of temperature and the mixture fraction. The clustering provided an initial partition of the curves with much less clutter than in Figure 3a. With the improved visualization, users can label distinct curve patterns and refine the categorization using domain knowledge. For more on these images, see the section “Clustering of attribute evolution curves” on p. 29.

branches. In the next section, we show how we used the cluster-label-classify strategy to confirm or deny these expectations and to qualify the nature of the transition.

Clustering of attribute evolution curves. In the phase space, we first used model-based clustering to partition the attribute evolution curves into an estimated number of groups. We knew that there were at least two distinct groups of correlation curves (the mixing and burning branches) and some others between them. So, we set the initial number of clusters at 10, a comparatively large number, in the hope of disclosing component patterns thoroughly. Naming an even larger number wouldn't have hurt the final results because we could have revised this parameter in the next step.

This automatic clustering aimed to alleviate the visual clutter in a single visualization (see Figure 3a) and reveal interesting cluster patterns (see Figure 4). Generally, these initial clustering results might not be the most satisfying. For example, Figures 4e, 4f, and 4g have similar patterns and would be better if they were combined. In addition, the clusters of Figures 4b and 4i contain outliers.

Semisupervised classification of attribute evolution curves. When analyzing attribute evolution curves, domain experts usually have certain background knowledge and might suggest their preferences in classifying the data. In this case study, two clusters of lines we wanted to see are those in Figures 5b and 5d, which correspond to the burning and mixing branches.

Figures 5a, 5c, 5e, and 5f depict sets of abnormal curves. The domain experts are very interested in such curves, which aren't clearly understood yet. So, we labeled representative curves in each of the six groups and divided the whole dataset into six classes, each with a distinct curve pattern.

Line data visualization and analysis. For each pair of images in Figure 6, the left image shows the classification results, and the right image shows the corresponding particle trajectories. The trajectory groups demonstrate that particles with different patterns of attribute evolution curves traversed the physical space differently. This process produced a much more reasonable and organized clustering.

OH and the Mixture Fraction

As in the previous case study, directly rendering all the attribute evolution curves in the phase space generated visual clutter (see Figure 7a). Also, as in that case study, a mixing branch and a burning

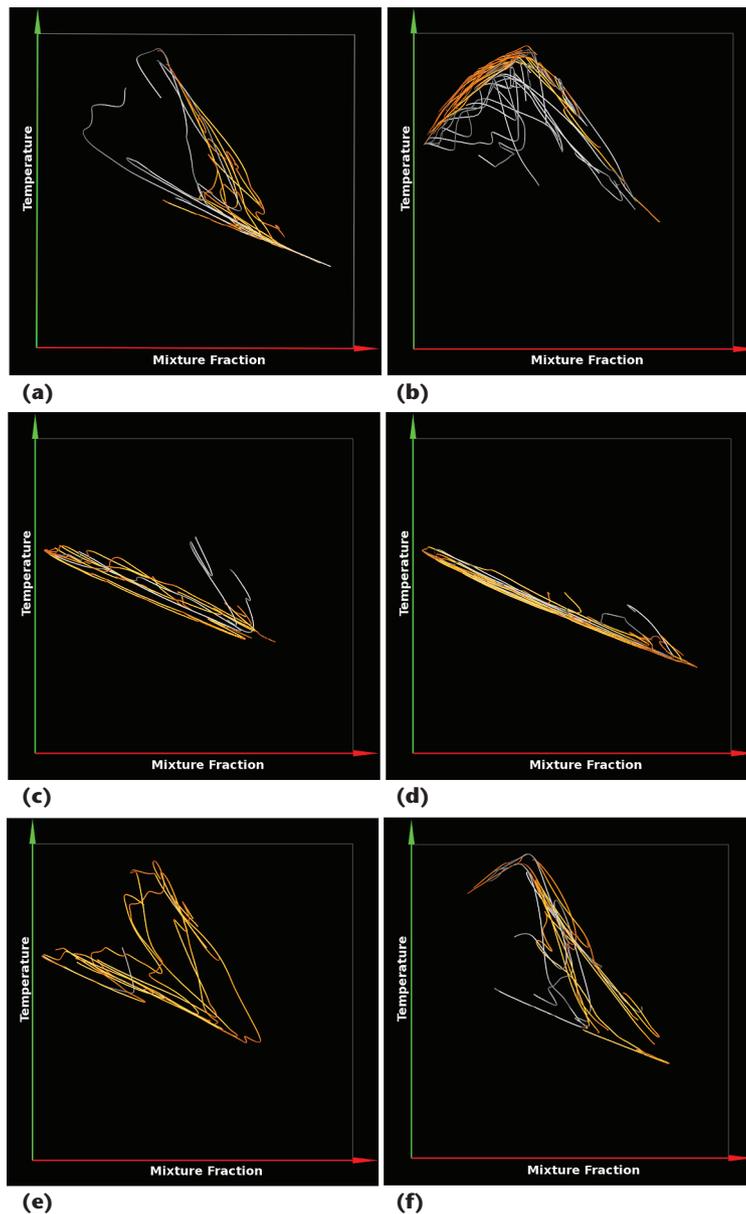


Figure 5. Six groups of curves correlating temperature and the mixture fraction, which the user labeled for semisupervised classification. These images correspond to Figures 4a, 4b, 4d, 4g, 4h, and 4i; the user employed our mouse-based picking tool to reject the outlier curves (in gray). For more on these images, see the section "Semisupervised classification of attribute evolution curves" on this page.

branch are discernible. The mixing branch is characterized by zero OH mass fraction (a horizontal line) for all mixture fractions. The burning branch has a positive correlation for low mixture fraction values and a negative correlation for high values. The mixing branch corresponds to pure mixing between the fuel and oxidizer with no chemical reactions, and hence the absence of radicals such as OH.

Clustering of attribute evolution curves. On the basis of our experience with the first case study, we first

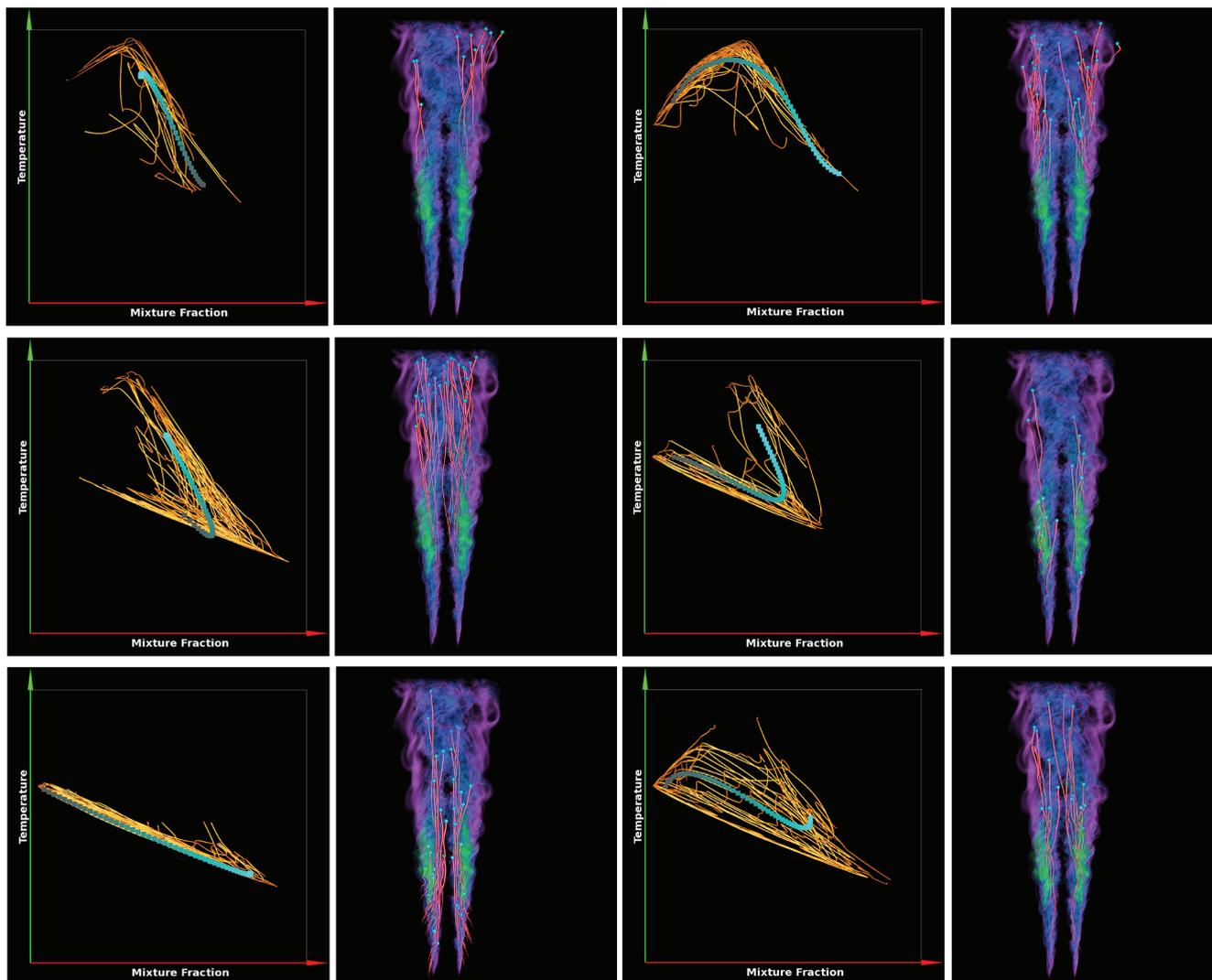


Figure 6. The classification result of curves correlating temperature and the mixture fraction. In each pair of images, the left image shows the classification results based on the user-labeled representative curves in each group. The dots, calculated with the regression mixture models, represent the average trends of classes. The direction is from the gray dot to the cyan dot. The right image in each pair shows the corresponding particle trajectories. Particles with distinct patterns of attribute evolution curves traversed the physical space differently.

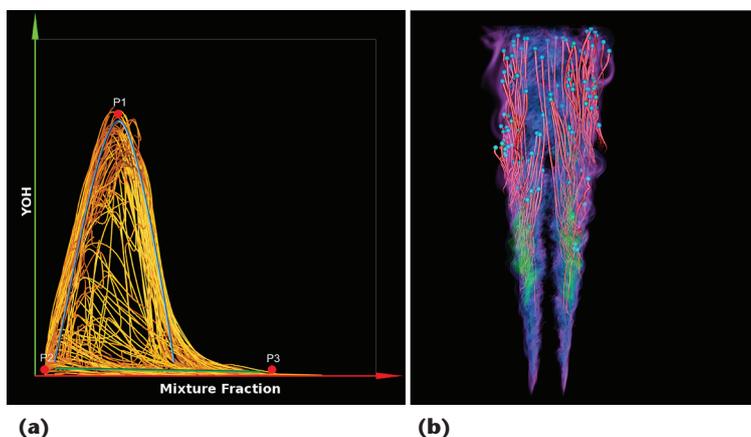


Figure 7. The relation between the hydroxide (OH) species and the mixture fraction. (a) The attribute evolution curves. (b) The corresponding particle trajectories, with volume rendering of the hydroperoxy field. Directly rendering all the attribute evolution curves generated visual clutter.

clustered all attribute evolution curves into 10 groups. The initial results in Figure 8 show distinct line patterns. For example, Figures 8b, 8d, and 8f have similar patterns and would be better if they were combined, as is the case for 8e and 8i. In addition, all the clusters contain outliers.

Semisupervised classification of attribute evolution curves. The preliminary clustering results revealed six major line patterns (see Figure 9). With the brushing and picking tools, we obtained six groups of representative lines for further classification.

Line data visualization and analysis. Figure 10 shows the classification results and the corresponding particle trajectories. The first four classifications are similar and seem to qualitatively resemble

the burning branch described earlier—a positive correlation for small values of the mixture fraction and a negative correlation for high values. On the other hand, the fifth classification seems close to the mixing branch characterized by nearly zero values of the OH mass fraction.

In the lifted-flame data considered here, chemical reactions occurred at a certain height, called the lift-off height, above the inlet. This region is marked approximately by the appearance of the green region in the volume rendering of hydroperoxy. All the trajectories for the first four classifications seem to originate above the lift-off height and thus represent burning solutions. In contrast, the trajectories for the fifth classification seem to originate closer to the inlet before liftoff and thus didn't experience much chemical activity. It's encouraging that the classification can extract these characteristics.

The sixth classification in Figure 10 is an interesting case; it seems to follow the mixing branch up to some point and then traverse the burning branch, but in the negative mixture fraction direction. Identifying such trajectories from large data would be difficult without our system. This classification, if statistically significant, is a valuable insight for domain experts from a modeling viewpoint.

Discussion

The combustion and visualization experts have been working closely together to develop and deploy the capabilities described in this article. There's an "unstimulated need" for this capability: the domain experts were struggling to process the actual data used to demonstrate the methodology in this article. When this project began, the domain experts had several hypotheses about the nature of the particle trajectories. However, they couldn't determine whether the particle data were consistent with their expectations, and they couldn't present the data coherently to the combustion community.

Using the current system, we've made an expository movie that the domain scientists have been using when discussing their simulation results with colleagues. (You can view the movie at <http://doi.ieeecomputersociety.org/10.1109/MCG.2011.108>.) Moreover, as we mentioned before, the dual-space technique can highlight the relationship between the particle attribute evolution curves and the particle trajectories. This provides combustion scientists with detailed information regarding the evolution of fluid parcels traversing a turbulent autoignitive environment.

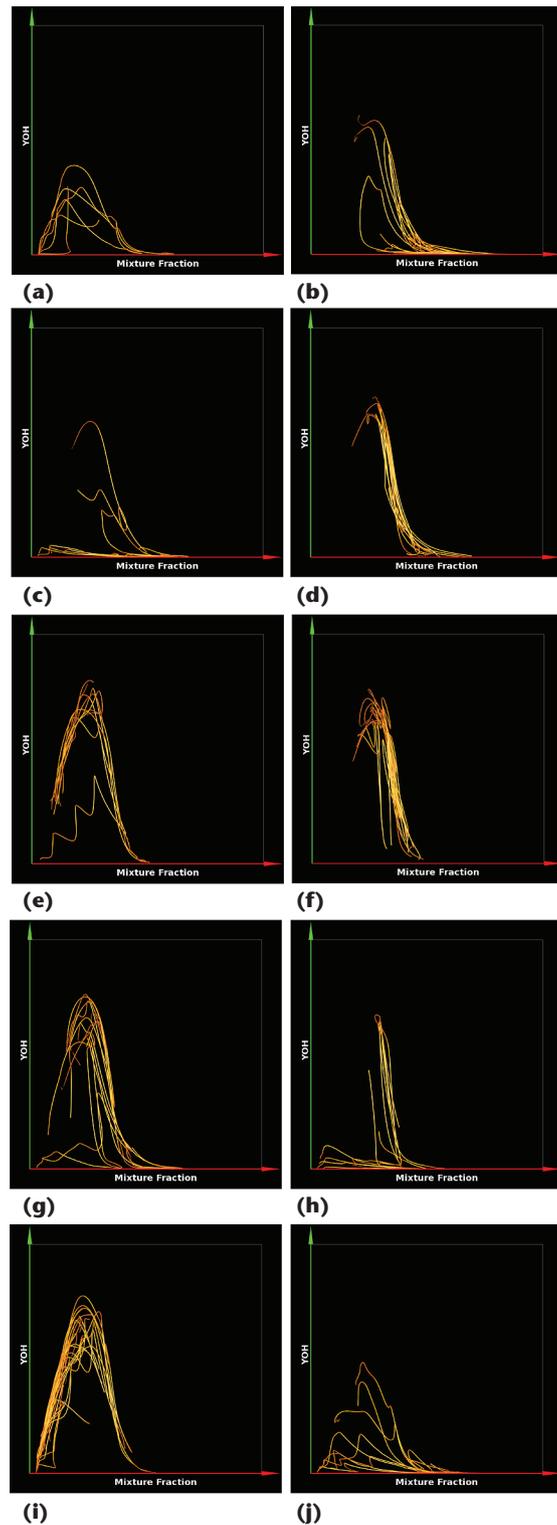


Figure 8. Automatic model-based clustering generated these 10 groups of attribute evolution curves of OH and the mixture fraction. The results provided an initial partition of the curves with much less clutter than in Figure 7a. With the improved visualization, users can label distinct curve patterns and refine the categorization using domain knowledge. For more on these images, see the section "Clustering of attribute evolution curves" on pp. 29–30.

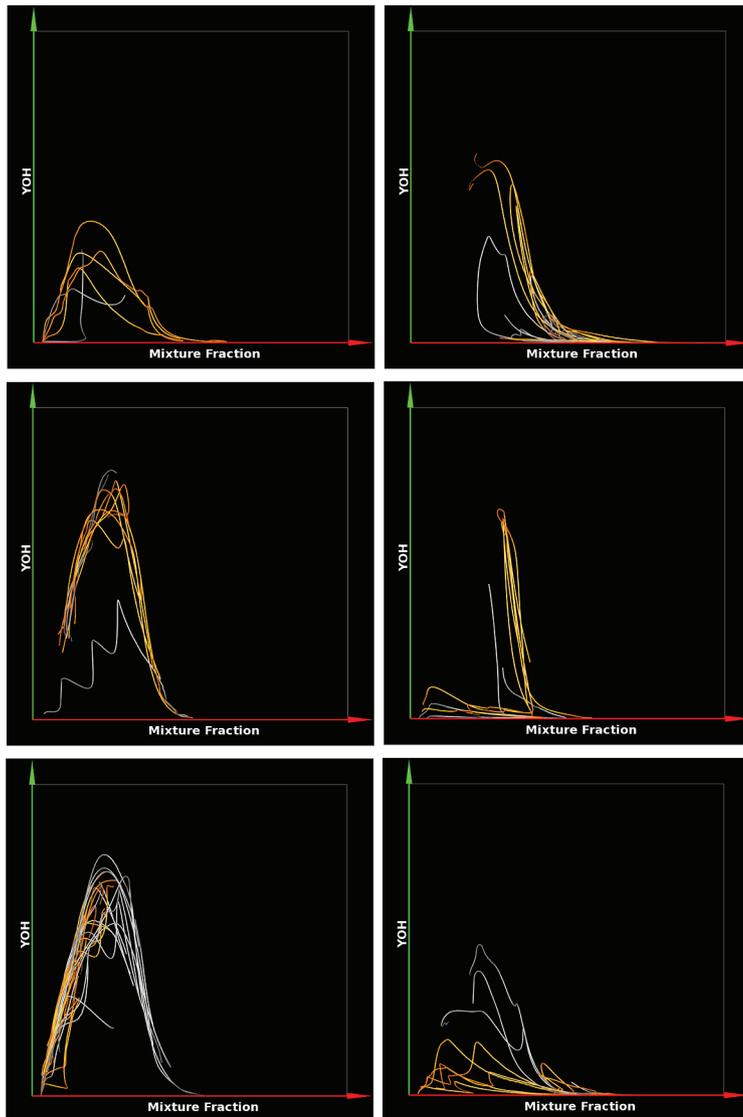


Figure 9. Six groups of curves correlating OH and the mixture fraction, which the user selected as prototypes for further semisupervised classification. These images correspond to Figures 8a, 8b, 8e, 8h, 8i, and 8j; the user employed our interaction tools to reject the outlier curves (in gray).

Advanced supercomputing continues to increase scientists' ability to model more complex problems at higher fidelity. Although our research targets turbulent-combustion simulations, model-based clustering and dual-space visualization are applicable to many other time-varying flow field data. The cluster-label-classify strategy is particularly powerful for analyzing large complex data that would be too dense and cluttered to visualize directly and wholly.

Interaction is the key to data analysis tasks, for not only exploring visual results but also steering semisupervised classification. We plan to improve our interaction tools and develop new ones to help users exploit their domain knowledge to steer

semisupervised classification. For instance, free sketching is a promising method by which users can specify curve patterns according to their knowledge. These curve patterns can then guide classification algorithms.

Currently, we use mixture models to analyze bivariate time series data. We could easily extend our method to handle multivariate time series data. But how to visualize and interactively explore the clustering or classification results of multivariate time series curves needs further study. So, we'll develop visualization and interaction methods to represent and manipulate high-dimensional clustering and classification results. Recently, we developed parallelized regression-mixture-model-based clustering that leverages the power of heterogeneous computers to categorize and visualize large line data derived from detailed scientific simulations.⁵ We plan to further investigate using parallel computing to improve our method's scalability with large simulation data. 

Acknowledgments

This research has been sponsored partly by the US Department of Energy (DOE) through the SciDAC (Scientific Discovery through Advanced Computing) program through agreement DE-FC02-06ER25777 and by the US National Science Foundation through grants OCI-0749227, CCF-0811422, OCI-0749217, OCI-0950008, and OCI-0850566. Sandia National Laboratories is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the DOE under contract DE-AC04-94-AL85000. We thank Hemanth Kolla for his valuable verification of our clustering results.

References

1. X. Zhu, *Semi-supervised Learning Literature Survey*, tech. report 1530, Computer Sciences, Univ. Wisconsin-Madison, 2005.
2. J. Wei et al., "Dual Space Analysis of Turbulent Combustion Particle Data," *Proc. 2011 IEEE Pacific Visualization Symp. (PacificVis 11)*, IEEE Press, 2011, pp. 91-98.
3. A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statistical Soc., Series B*, vol. 39, no. 1, 1977, pp. 1-38.
4. N. Peters, "Laminar Diffusion Flamelet Models in Non-premixed Turbulent Combustion," *Progress in Energy and Combustion Science*, vol. 10, 1984, pp. 319-339.

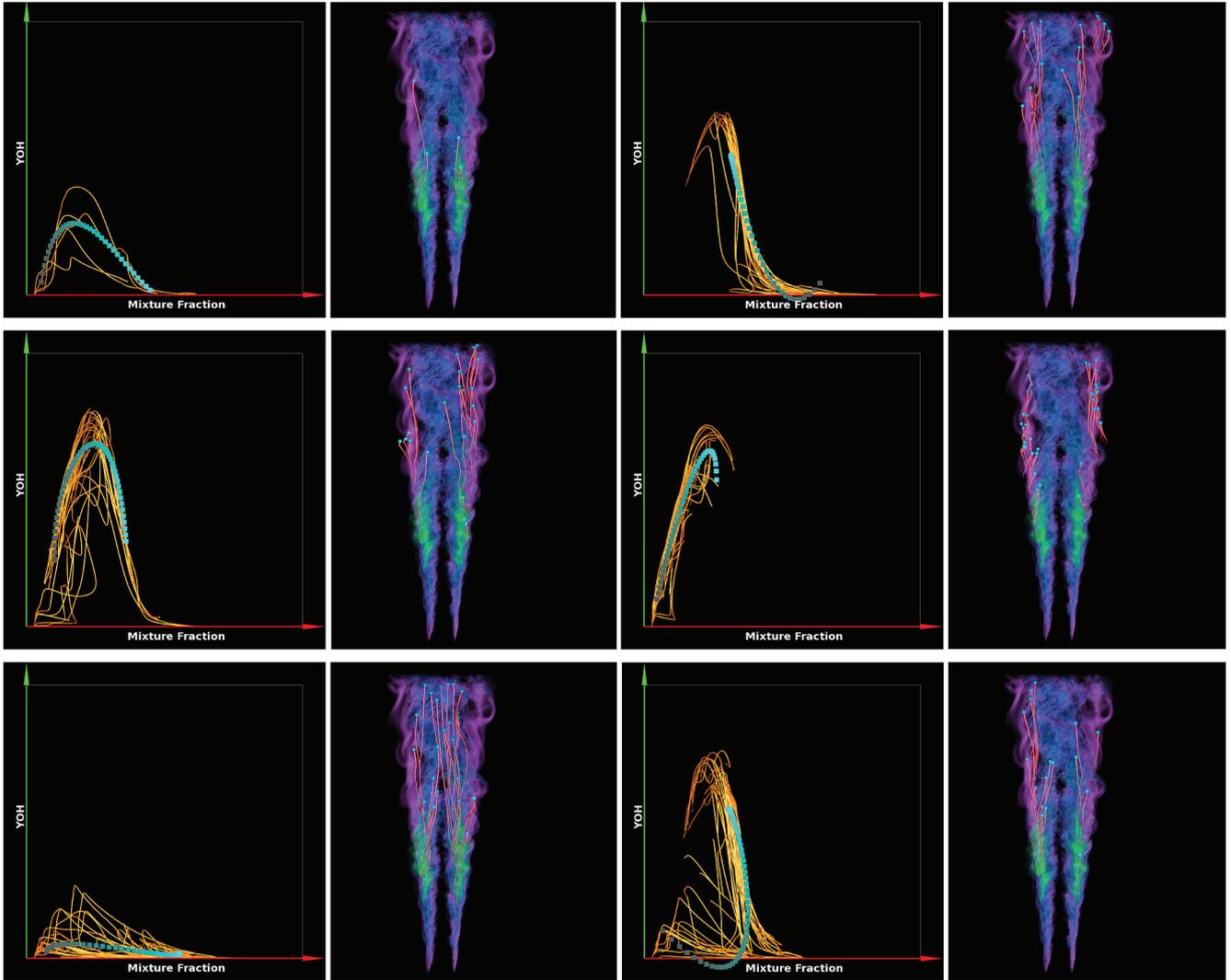


Figure 10. The classification result of curves correlating OH and the mixture fraction. In each pair of images, the left image shows the classification results based on the user-specified class prototypes. The dots, calculated with the regression mixture models, represent the average trends of classes. The direction is from the gray dot to the cyan dot. The original OH data were all positive values, but some dots on the representative curves go below the x -axis. This is because we approximated the original lines with B-spline models during preprocessing. The right image in each pair shows the corresponding particle trajectories. Particles with distinct patterns of attribute evolution curves traversed the physical space differently.

5. J. Wei et al., "Parallel Clustering for Visualizing Large Scientific Line Data," *Proc. IEEE Symp. Large Data Analysis and Visualization (LDAV 11)*, IEEE Press, 2011, pp. 68–76.

Jishang Wei is a PhD candidate in computer science at the University of California, Davis. Contact him at jswei@ucdavis.edu.

Hongfeng Yu is a postdoctoral researcher at Sandia National Laboratories. Contact him at hyu@sandia.gov.

Ray W. Grout is a high-performance-computing applications researcher at the National Renewable Energy Laboratory. Contact him at ray.grout@nrel.gov.

Jacqueline H. Chen is a distinguished member of the technical staff at Sandia National Laboratories, an adjunct professor of chemical engineering at the University of Utah, and a member of the Combustion Institute Board of Directors. Contact her at jhchen@sandia.gov.

Kwan-Liu Ma is a professor of computer science and the chair of the Graduate Group in Computer Science at the University of California, Davis. Contact him at ma@cs.ucdavis.edu.



Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.