A CONFIDENCE-GUIDED TECHNIQUE FOR TRACKING TIME-VARYING FEATURES

Soumya Dutta
Information Sciences, Los Alamos National Laboratory

Terece L. Turton
Information Sciences, Los Alamos National Laboratory

James Ahrens
IS&T Institute, Los Alamos National Laboratory

***Abstract*—Application scientists often employ feature tracking algorithms to capture the temporal evolution of various features in their simulation data. However, as the complexity of the scientific features is increasing with the advanced simulation modeling techniques, quantification of reliability of the feature tracking algorithms is becoming important. One of the desired requirements for any robust feature tracking algorithm is to estimate its confidence during each tracking step so that the results obtained can be interpreted without any ambiguity. To address this, we develop a confidence-guided feature tracking algorithm that allows reliable tracking of user-selected features and presents the tracking dynamics using a graph-based visualization along with the spatial visualization of the tracked feature. The efficacy of the proposed method is demonstrated by applying it to two scientific data sets containing different types of time-varying features.**

IN THE ERA of big data analytics, experts regularly use feature tracking algorithms to effectively explore time-varying data. However, complex scientific features undergo evolutionary events that pose significant challenges for their robust tracking. To solve the *correspondence problem* in tracking, researchers have proposed several techniques [1], [2], [3], demonstrating their usefulness in various scenarios. Most of these techniques have not focused on studying the reliability of these techniques. As a result, if a tracking algorithm makes an incorrect correspondence, the users cannot determine there is an error unless every tracking step is manually investigated – a time-consuming process. Therefore, a key requirement for any tracking algorithm is to report its confidence and inform the users about the existing uncertainty.

 Published by the IEEE Computer Society

Most of the previous feature tracking works can be broadly classified into two categories: (1) attribute-based tracking [2], [4], [5], and (2) volume overlap-based tracking [3], [6], [7]. Both of these categories have shown promising results. For attribute-based methods, feature correspondence is established by comparing several feature attribute values with a set of predefined fixed threshold values. Optimal threshold values can be difficult to determine yet the effectiveness of the algorithm relies heavily on them. Different data sets require different thresholds based on their temporal dynamics. The volume-overlap based techniques require high temporal resolution and are not applicable when two corresponding features do not overlap in time [1]. Given their respective limitations, an uncertainty-aware feature tracking algorithm can enhance the robustness of both of these classes.

We present a new confidence-guided feature tracking algorithm that overcomes the potential limitations of both the attribute and overlap-based tracking algorithms and increases their robustness. Given several feature attributes, a *fuzzy rule based system* (FRBS) captures their temporal dynamics and estimates various feature behaviors using a set of fuzzy rules. The target feature is then fed into the FRBS for inference-based tracking. The FRBS solves the correspondence in future time steps using its learned knowledge-base, estimating a confidence score for each correspondence testing. The proposed system is trained to detect the continuing features with high confidence. When an unexpected event such as a feature split or merge occurs, the confidence score of the fuzzy system becomes low indicating high tracking uncertainty and the occurrence of an evolutionary event. Using the proposed method, such time steps are readily identified and further investigated to categorize the detected event. The proposed method has several unique advantages. Firstly, the method does not require a set of predefined attribute thresholds for feature correspondence checking. Instead, a consistent and interpretable confidence score is generated. This score indicates the feature matching confidence, enhancing the overall robustness of the tracking algorithm. Secondly, the attribute similarity measures are used to detect feature correspondence rather than overlap criterion, making the method applicable for temporally sparse data sets.

The effectiveness of this method is demonstrated with two scientific data sets and with sparsely time sampled use cases. Our contributions in this work are thus twofold: **(1)** A knowledge-driven fuzzy rule based algorithm capable of tracking dynamic features and quantifying the feature matching confidence at each step, and **(2)** Visualization of the important tracked features over time with a new confidence-guided tracking graph to convey the overall tracking dynamics to scientists.

RELATED WORKS

Feature tracking is an important task in scientific data visualization. Samataney et al. [2] proposed an attribute-based correspondence approach to track volume features in scientific data sets. Reinder et al. [4] introduced a similar attribute-based feature tracking algorithm. Silver and Wang [3] tracked features by exploiting volume overlap criteria. Ji and Shen used the earth mover's distance to design an optimum feature tracking algorithm [1]. Using a predictor-corrector method, Muelder and Ma introduced a new algorithm for efficient feature tracking [8]. Dutta and Shen [5] proposed feature tracking using distribution-based data sets. Saikia and Weinkauf introduced a global feature tracking algorithm where feature correspondence was measured using volume overlap and distribution differences [7]. Schnorr et al. [9] introduced a two-step optimization algorithm for feature tracking.

KNOWLEDGE-DRIVEN TRACKING

This work makes use of a fuzzy rule based system (FRBS) to quantify the tracking confidence. The FRBS produces a confidence score at each step for the match so users can judge tracking reliability. When a sudden evolutionary event, e.g., a feature split/merge, the matching confidence score drops significantly, prompting further user attention. Visual exploration of the tracking results is conducted by volume visualization and via a new tracking graph depicting the overall tracking dynamics. Note that feature extraction is not covered in this work and the methodology assumes features can be extracted using any appropriate feature extraction algorithm.

Given a target feature f_i and a set of extracted candidate objects $O = \{O_1, O_2, ..., O_k\}$, the goal of a tracking algorithm is to identify the object from the set O which correctly corresponds to the target feature. Traditional attribute-based feature correspondence detection in this context has been shown to be promising [2], [4], [5]. The correspondence criteria is measured by computing the differences between several attribute values and then checking the differences against predefined hard thresholds. The best match is determined by picking the closest object satisfying all the threshold conditions. If no match is found, a feature dissipation/death event is indicated. A potential drawback of this technique is that it relies upon multiple hard thresholds. These thresholds are generally: (a) set manually, (b) depend on the data set and feature dynamics. Determining a consistent and robust set of thresholds is non-trivial, often requiring expert tuning.

ESTIMATION OF TEMPORAL FEATURE DYNAMICS

We propose a new knowledge-driven fuzzy rule based tracking algorithm that first captures the feature dynamics from the temporal evolution of representative features and then uses the acquired knowledge to track other features. The motivation of using a fuzzy rule based system to address this problem are twofold: (1) a fuzzy rule based system provides an effective way to map the attribute based correspondence detection problem to a rule based system without explicitly specifying any hard thresholds; and (2) the working principle of the fuzzy learning algorithm is well understood, reducing the impact of model uncertainty while analyzing the results. To compactly model the dynamic behaviors of the features, we use a set of fuzzy rules where each rule models a specific behavioral pattern of the feature dynamics. To capture the temporal dynamics of a feature, we compute four key attributes for each object: (1) mass (M), (2) volume (Vol), (3) centroid (C), and (4) velocity (Vel). These quantities are computed via the methodology described in [2], [4]. Besides these four attributes, other feature attributes related to feature shape, orientation, and higher order moments can also be added to the fuzzy analysis system.

FUZZIFICATION OF FEATURE ATTRIBUTE SIMILARITIES Given a set of candidate objects, the fuzzy system aims to produce the highest output response for the true corresponding object. The similarity between a candidate object and the target feature can be estimated by measuring the difference in their attribute values. These differences are then used as the input to the fuzzy system. Conceptually, the smaller the differences are, the higher the similarity between the object and the target feature is, and hence, a higher confidence output is desired. Since the fuzzy system works in a fuzzy domain where the input is mapped to a fuzzy value, a transformation of the attribute difference values into the fuzzy domain is necessary. A consistent and comparable representation of the difference values in the fuzzy domain is achieved using membership functions which quantify the degree of attribute differences to a fuzzy value $\in [0,1]$. This method is known as *fuzzification*. Note that we can quantify the attribute similarity criteria during tracking without requiring predefined thresholds. In order to define the fuzzification process, we use the Gaussian membership function (GMF). Note that the GMF is one approach possible from finite mixture models (see e.g., [10]) and other types of membership functions can also be used. Given x, an attribute difference value, with $\bar{x}$ as its mean, and σ as its standard deviation, $\Delta x = x - \bar{x}$. The GMF is formally defined as:

$$GMF(x) = exp(-\Delta x^2 / 2\sigma^2) \qquad (1)$$

If, for example, ΔVol represents the difference in volume attribute for an object when compared with the volume of the target feature, then by using a GMF, ΔVol gets mapped to a fuzzy value, reflecting the degree of similarity to its associated target feature volume.

CONSTRUCTION OF KNOWLEDGE-BASE FOR TRACKING To build the knowledge-base for tracking, we employ a fuzzy clustering-based learning scheme. The purpose of this clustering-based learning is to extract the natural groupings from the known training data which can be used to concisely depict the behaviors of the features in the data in terms of several fuzzy rules [11]. The correspondence check between a candidate object and the target feature is done using an attribute vector containing the attribute difference values. Thus a vector of the form $\{\Delta\text{Vel}, \Delta\text{M}, \Delta\text{Vol}, \Delta\text{C}\}$ becomes an input for the fuzzy system.The final output from the fuzzy system is a scalar response value. The goal is to produce a high response if all difference values are small, indicating that the candidate object is very similar to the target feature. These criteria associate a high possibility for that candidate object to be the true corresponding feature.

Generation of training data.
To capture the dynamic pattern from attribute value differences, we first create training data that conforms with the correct tracking results and then use it to learn the parameters of the fuzzy system. During training, the system will first learn the parameters for the Gaussian membership functions. To model the output function, a least square estimation (LSE) is required. The coefficients of the LSE will be learned using the training data.

Several representative features are selected from the data and manually tracked over time. While labeling the features, all segmented features are visualized for each time step and the correct corresponding feature can be easily selected by visually inspecting them. During the labeling process, each representative feature is tracked for a span of 10-15 time steps to collect the desired training data. Note that there is a set of candidate objects at each time step and only one of them gives the correct correspondence. We measure the 4 attributes $\{\Delta\text{Vel},\Delta\text{M},\Delta\text{Vol},\Delta\text{C}\}$ for each of the candidate object at each time step. Since we know the ground-truth feature, we assign a high response value (0.9) to the output variable for the correct corresponding feature, and a low response (0.1) is set to the output variable for all the other objects. This results in a 5D labeled training data (the 4 attribute components plus the assigned scalar response).

Estimation of Parameters for the GMFs.
Given this 5D labeled training data, it can be grouped into several clusters with each cluster modeling a specific behavioral pattern of the features. For example, a cluster where all the input attributes are very low and the output is high will represent the group of objects with a very high chance of being the target feature. Such a cluster can be formally modeled as a fuzzy rule in the form: *IF (antecedent) THEN (consequent)*. For our feature tracking application, such a predicate-based fuzzy rule can be written as: IF (ΔVel is LOW AND ΔM is LOW AND ΔVol is LOW AND ΔC is LOW) THEN output is HIGH. Similarly, another cluster where the input values are high and the output value is low can be translated as: IF (ΔVel is HIGH AND ΔM is HIGH AND ΔVol is HIGH AND ΔC is HIGH) THEN output is LOW. To extract such dynamic rules, we apply a fuzzy-C-means (FCM) clustering algorithm to this 5D training data first. The efficiency of this FCM in extracting the fuzzy clusters has been demonstrated in [12]. Given $\mathscr{X} = \{x_1, x_2, ..x_n\}$, ($n$ is the number of data points) as the input to the FCM, the algorithm produces a set of centroids $\mathscr{V} = \{v_1, v_2, ..v_c\}$ (c is the number of clusters) and a membership matrix $\mathscr{M}$ of dimension $c \times n$ by minimizing an objective function [11]. An element of m_{ik} of this membership matrix represents the membership value of kth data point in ith cluster.

Each cluster center obtained from the FCM becomes a representative of one of the feature's behavioral patterns as described above. The degree of fulfillment for each of the sub-clause in a rule of the form (ΔVol is LOW) is estimated by its associated GMF. The estimated centroids for each cluster obtained from the FCM become the suitable choice for means of the corresponding GMFs, and the standard deviation of each GMF is computed as suggested in [11], [12].

CONFIDENCE GUIDED FEATURE TRACKING

Given the GMFs, we discuss the inference technique for the fuzzy system and how new features can be tracked using it. Here we adopt the widely used *Takagi-Sugeno fuzzy rule based system* (TS-FRBS), which has been shown to be effective in modeling dynamic systems [11]. The output response in a TS-FRBS is modeled as a linear function of input variables. The value of this output indicates the confidence of the system for the input tested. Formally, given a specific input attribute vector $(x_1, ..., x_q)$, and a set of fuzzy rules R^j, $\quad (j = 1, 2, ..c)$ where c is the number of rules and q is the number of attributes, the output is inferred as follows. First, the input is tested with each of the rules and a degree of

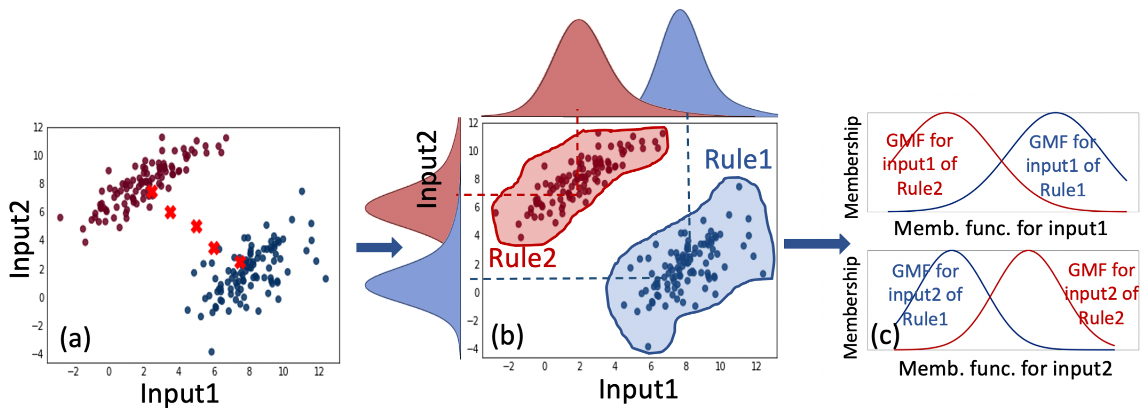

Figure 1. Demonstration of the fuzzy rule estimation scheme using a 2D bivariate data. Two rules are generated and each color shows one rule. Five test points from Table 1 are shown using red crosses in Figure 1a.

match is computed, called the *firing strength* of that rule for the input. The firing strength α^j of jth rule is computed as:

$$\alpha^j = GMF_1^j x_1^j \ \wedge \ GMF_2^j x_2^j \ \dots \ \wedge \ GMF_q^j x_q^j \quad (2)$$

where $GMF_1^j, GMF_2^j \ \dots \ GMF_q^j$ are the GMFs of the form described in Equation 1 for the jth rule, $\wedge$ is the fuzzy T-norm conjunction operator [11], and we have used the multiplication as our conjunction operator to combine the sub-clauses in each rule. The firing strength intuitively estimates the degree of match of rule R^j for the given input by combining the contribution coming from each feature attribute clause using the fuzzy conjunction operator. So, if most of the clauses in the input have satisfied strongly in a rule, then the firing strength of that rule for the input will be high. Now, since the output variable y^j is a linear function of the input variables, so the output function $\psi(\cdot)$ can be represented as:

$$o^j = \psi(x_1^j, \dots, x_q^j) = \beta_0^j + \beta_1^j \cdot x_1^j + \dots + \beta_q^j \cdot x_q^j \quad (3)$$

where $\beta_0^j, \beta_1^j \dots \beta_q^j$ are the coefficients of the linear function $\psi(\cdot)$. Then the final output response $\mathscr{O}$, inferred from c rules for a specific input $x_1, \dots, x_q$, is given as the average of all o^j values weighted by their firing strengths and can be expressed as:

$$\mathscr{O} = (\sum_{j=1}^{c} \alpha^j \cdot o^j)/(\sum_{j=1}^{c} \alpha^j) \qquad (4)$$

Given the GMFs parameters and the training data generated from the attribute difference values, the parameters $\beta_0^j, \beta_1^j \dots, \beta_q^j$ are computed by optimization with respect to the training data and the optimization reduces to a linear least square estimation problem as described in [12]. So, at every time step, the extracted candidate objects can be evaluated using this TS-FRBS, and the object that produces maximum output response is identified as the corresponding continuing feature and this process is repeated over time. Algorithm 1 provides a detailed pseudo code for the proposed tracking algorithm.

ILLUSTRATION OF THE FUZZY SYSTEM USING SYNTHETIC DATA

In Figure 1, we illustrate working principles of this fuzzy system using a synthetic bi-variate 2D data set obtained by randomly sampling points (Figure 1a) from two 2D multivariate Gaussian distributions centered at (2,8) and (8,2) respectively. The output value for points generated from the Gaussian centered at (2,8) or (8,2) was set to 0.0 or 1.0, respectively. This resulted in a 3-tuple training data, and two clusters, i.e., two fuzzy rules were constructed. Each color represents one fuzzy rule in Figure 1c.

Table 1 shows testing results of five test points to demonstrate the algorithm functionality. These points are selected such that the first point is close to (2,8) and the last point is close to (8,2), with the remaining points in between. We see that the output for the first point is 0.1096, close to zero, while for the fifth point, the output is close to 1. The point which is equally distant from the two cluster centers produces an output of 0.5136. This point can be considered as an uncertain observation since the fuzzy system was not able to produce a high confidence. For cases like these, when a hard classification is not suitable, the use of a fuzzy system allows us to make a confidence-driven decision.

Algorithm 1: TS-FRBS inference-based feature tracking algorithm with uncertainty estimation.

Output1: List of all tracked features from all time steps.
Output2: List of all confidence scores from all time steps for all objects.
Initialization:
start time = t_0
end time = t_n
conf_TH = User specified confidence value
tf = User selected target feature to track
for $t \leftarrow t_0$ **to** t_n **do**
  $\{o_1, o_2, .., o_k\}$ = Extracted candidate objects at time t using an existing feature extraction algorithm.
  confidence_scores = []
  for $o_i \leftarrow o_1$ **to** o_k **do**
    $o_i^{att\_diff}$ = $Comp\_attr\_diff(o_i,\ tf)$
    c = $TSFRBS(o_i^{att\_diff}, tf)$
    confidence_scores.append(c)
  max_score, matched_fid = $Find\_best\_match$(confidence_scores)
  if $max\_score < conf\_TH$ **then**
    Investigate the time step for evolutionary events.
  else
    tf = feature(matched_fid)
    Continue tracking tf to next time step.

RESULTS

While generating the rule based systems, the number of rules are often chosen based on application needs [12]. In our case, we found that $3 \sim 5$ rules are generally sufficient to capture the feature dynamics. We chose to use 3 rules for both data sets to obtain consistent results. A threshold is set on the confidence value to flag features for potential merge/split events. This threshold was set to a high value of 0.7 for all experiments. When the confidence value dropped below 0.7, the features in those time steps were investigated further for evolutionary events. All the experiments were done on a MacBook Pro with a 3.1 GHz Quad-Core Intel Core *i*7 processor and 16 GB memory. Both training and testing code were run serially.

Table 1. Results of several test points to demonstrate the working of the fuzzy rule based system.

Point id	Input test point	Generated output response
1	(2.5, 7.5)	0.1096
2	(3.5, 6.0)	0.3084
3	(5.0, 5.0)	0.5136
4	(6.0, 3.5)	0.7125
5	(7.5,2.5)	0.9176

TRACKING IN VORTEX DATA SET

The Vortex data is a pseudo-spectral simulation of coherent vortex cores. The data set has a spatial resolution of $128 \times 128 \times 128$ and 30 time steps. The scalar variable is vorticity magnitude. The features are identified as segmented regions with segmentation criterion: scalar value ≥ 7.0 (high vorticity values). Each connected component is treated as a separate feature. The training data was generated by tracking a representative vortex manually over 10 time steps and the four attributes for each object were computed. For the correct corresponding feature, a confidence value of 0.9 was assigned and set to 0.1 for all the other objects.

The fuzzy system generated consists of 3 rules shown in Figure 2(left). The training set consisted of 88 data points and 10.48secs were needed to train the fuzzy system. Each rule is shown using a different color and has four GMFs, one corresponding to each feature attribute. We observe that Rule2 (blue), captures the low valued feature attribute differences, modeling objects similar to the target feature. Thus this rule will contribute the most when the final output will be computed using Equation 4. The other two rules (green and blue) will contribute for the objects which are not similar to the target feature and will produce a low confidence score.

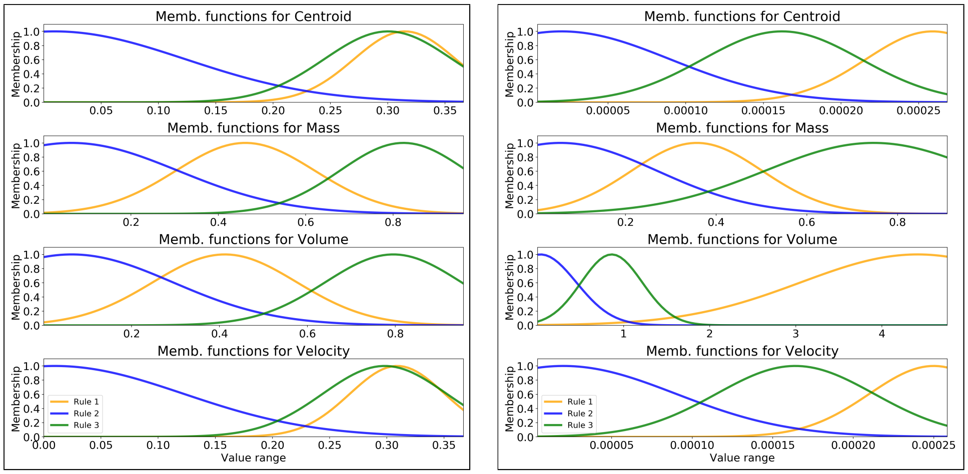

Figure 2. Membership functions for the Vortex data (left) and MFiX-Exa data (right). Each color represents one of the fuzzy rules.

Figure 3(top) shows the tracking results of a vortex feature. The tracking algorithm took 0.54 secs on average per time step to detect the correct corresponding feature. The tracking results were manually verified to ensure the correctness of the proposed method. Also, a video is provided that shows the tracking for all time steps. The tracking graph (top) shows the overall tracking dynamics. The target feature, selected at t=10, was tracked consistently for 15 time steps. In the tracking graph, extracted objects from each time step are stacked vertically and time steps are laid out horizontally. Each node represents an object at a specific time step and the node color and size shows the tracking confidence value inferred by the fuzzy system. The red line through the graph connects the correct corresponding object over time as it is being tracked reliably and the confidence score is the highest for this object at each time step. This graph also presents the matching confidence values for the other tested objects during tracking at each time step so that the overall reliability and the dynamics of the fuzzy tracking algorithm can be studied. At $t = 25$, the maximum confidence score was 0.643 ($<$ predefined threshold of 0.7) and so this time step was further investigated, with a split event found. Figure 3(bottom) shows the spatial volume visualization of the tracked feature (red) for three different time steps, along with all the other candidate objects present (green). Note that as the features change their position and shape over time, the proposed fuzzy system is able to track the target feature correctly with high confidence.

TRACKING IN MFIX-EXA DATA SET

Our second case study uses data generated from MFiX-Exa, a multi-phase flow simulation code (amrex-codes.github.io/MFIX-Exa) used to study reactions in fluidization beds of chemical looping reactors. In this study, we used the particle density field. An important phenomenon in this density data is the formation of bubbles, which generally reflect low density regions. A bubble detection algorithm identifies the bubble features as connected regions with very low density. The spatial resolution of the density field used is $128 \times 16 \times 128$. Since the simulation data changes slowly over time, the simulation data was stored at every 100th iteration to reduce overall storage, resulting in 408 time steps. Since our tracking method does not need the overlap criterion, it is well-suited for this sparsely time-sampled data set.

We selected two representative bubbles and tracked them manually to generate the training data. The fuzzy system with 3 rules generated is shown in Figure 2(right). The training set consisted of 90 points, taking 22.68 secs to train. As before, Rule2 (blue) captures the low valued feature attribute differences representing the objects that are very similar to the target feature.

We use these fuzzy rules to track bubbles in the data set and find that tracking a bubble takes

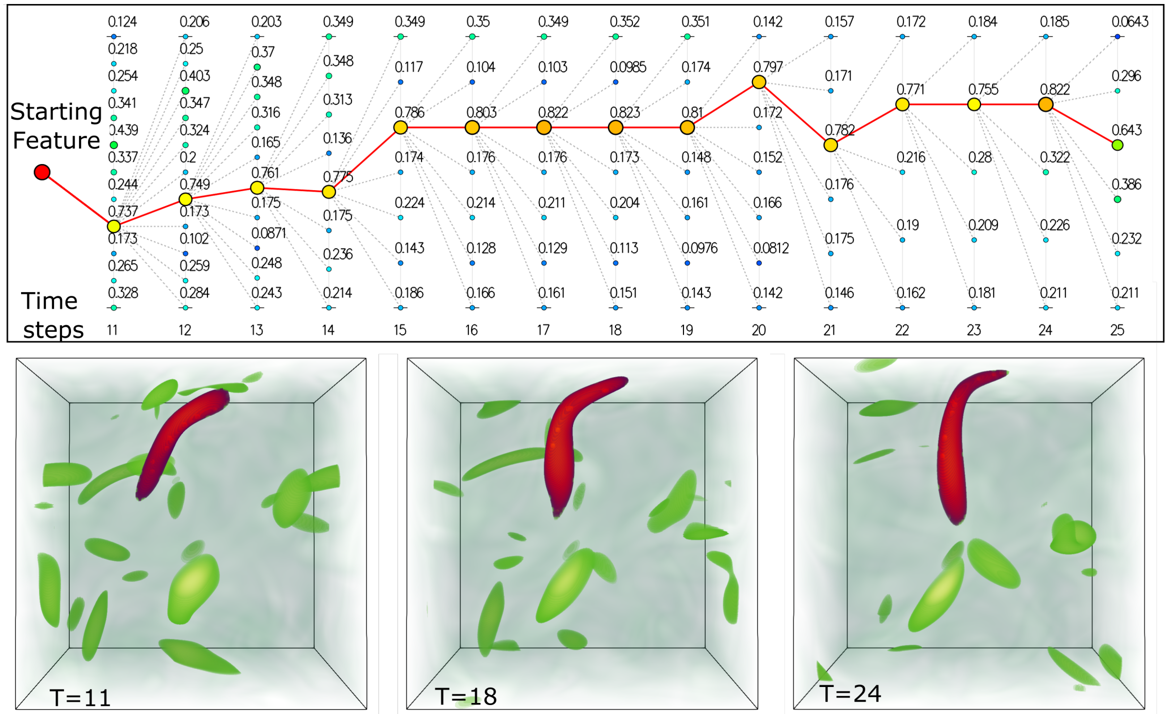

Figure 3. Feature tracking result for Vortex data set. The confidence-guided tracking graph is shown (top) for the tracked time window. Note at $t = 25$, the confidence value dropped below 0.7, indicating the occurrence of a feature split. The tracked feature is shown at three different time steps (bottom).

0.1225 secs on average per time step. To further study the effectiveness of the technique when only sparsely sampled time steps are available, we fed the system every 4th time step (i.e., every 400th simulation iteration snapshot). The resulting tracking graph for a selected bubble from time step 19300 is shown in Figure 4. The proposed technique tracked the bubble correctly even for sparsely sampled time steps. The spatial volume visualization of three representative time steps is shown in Figure 4(bottom). All the other bubble features are shown in black for context while the tracked bubble is highlighted in reddish-yellow. From the tracking graph, we observe that at t=23800, the confidence score drops below the confidence threshold (0.7). Further investigation reveals that the bubble merged with another bubble, causing the low confidence value.

DISCUSSION AND CONCLUSION

The key advantage of the proposed tracking algorithm is that it does not need a set of user-specified hard thresholds for detecting correspondence among feature attributes. Earlier attribute-based techniques relied upon a multi-threshold based correspondence detection. Our proposed method extends the robustness of such algorithms by removing the requirement of those preset thresholds and instead a consistent and interpretable confidence score is used across different data sets. We also showed that the proposed technique can be used on sparely time-sampled data sets and feature overlap is not required. However, we found that as time steps became more sparse, the fuzzy system made erroneous correspondence. We have tested the proposed system on two challenging data sets and demonstrated promising tracking results.

A potential limitation of our work is that the technique can only detect feature continuation events automatically with high confidence. For feature split/merges, the correct categorization of such events requires user interaction. Hence, in the future, we plan to extend our technique to detect those events automatically. Also, we plan to include more feature attributes in the system, explore other mixture models as the fuzzification basis, and study the generalizability of the pro-

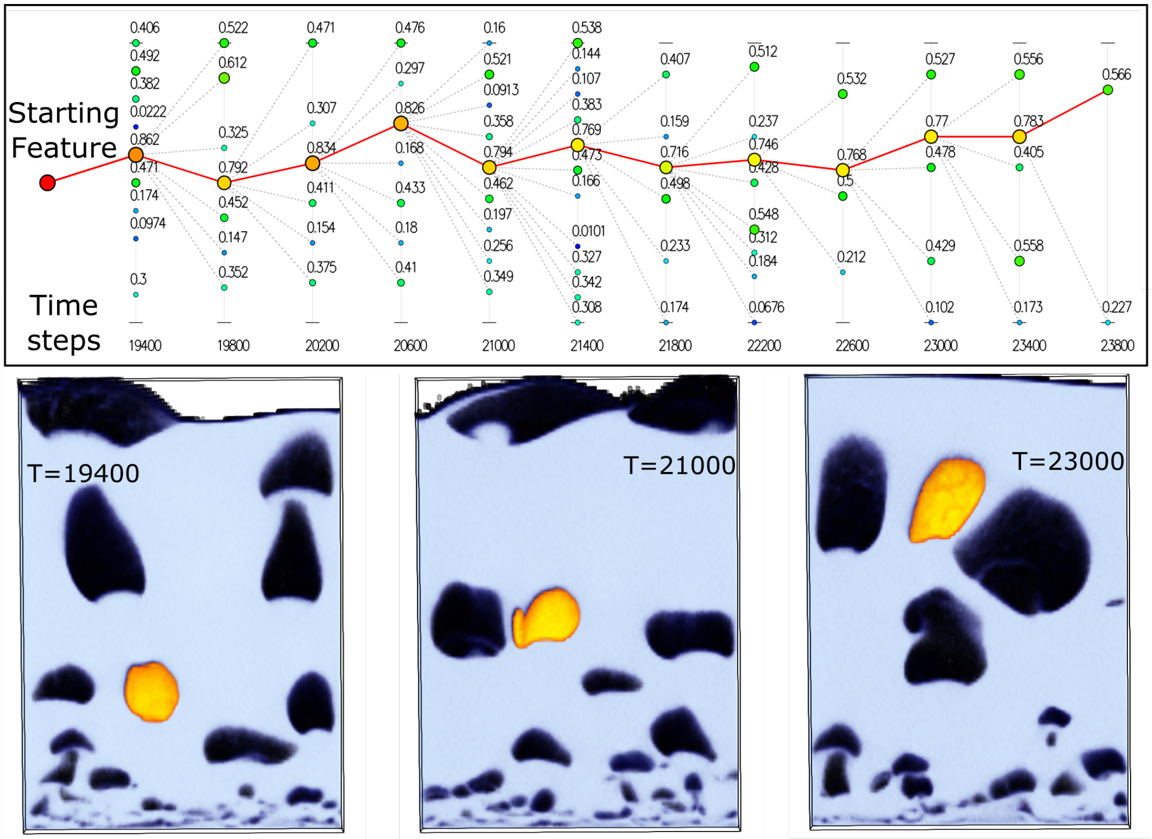

Figure 4. Top: Feature tracking result for the MFiX-Exa data set. The confidence-guided tracking graph is shown for the tracked time window (top). At t=23800, the confidence value dropped below 0.7 indicating the occurrence of a feature merge. Bottom: The evolution of the feature at three different time steps.

posed fuzzy system for tracking features in situ.

ACKNOWLEDGMENT

The authors would like to thank the Department of Energy and Los Alamos National Laboratory for the funding and support in carrying out this research. This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration and is released under LA-UR-20-28907.

REFERENCES

1. G. Ji and H.-W. Shen, “Feature tracking using earth mover’s distance and global optimization, pacific graphics,” 2006.
2. R. Samtaney, D. Silver, N. Zabusky, and J. Cao, “Visualizing features and tracking their evolution,” *Computer*, vol. 27, pp. 20–27, 1994.
3. D. Silver and X. Wang, “Volume tracking,” in *In Proceedings of the Visualization ’96 Conference*. Computer Society Press, 1996, pp. 157–164.
4. F. Reinders, F. H. Post, and H. J. W. Spoelder, “Attribute-based feature tracking,” in *Data Visualization ’99*. Springer Verlag, 1999, pp. 63–72.
5. S. Dutta and H. Shen, “Distribution driven extraction and tracking of features for time-varying data analysis,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 22, no. 1, pp. 837–846, Jan 2016.
6. D. Silver and X. Wang, “Tracking and visualizing turbulent 3d features,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 3, no. 2, pp. 129–141, Apr 1997.
7. H. Saikia and T. Weinkauf, “Global feature tracking and similarity estimation in time-dependent scalar fields,” *Computer Graphics Forum*, vol. 36, no. 3, pp. 1–11, 2017.
8. C. Muelder and K.-L. Ma, “Interactive feature extraction and tracking by utilizing region coherency,” in *Visual-*

ization Symposium, 2009. PacificVis '09. IEEE Pacific, April 2009, pp. 17–24.
9. A. Schnorr, D. N. Helmrich, D. Denker, T. W. Kuhlen, and B. Hentschel, "Feature tracking by two-step optimization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 6, pp. 2219–2233, 2020.
10. V. Melnykov and R. Maitra, "Finite mixture models and model-based clustering," *Statist. Surv.*, vol. 4, pp. 80–116, 2010.
11. T. Ross, *Fuzzy Logic with Engineering Applications*. Wiley, 2004.
12. N. Pal, V. Eluri, and G. Mandal, "Fuzzy logic approaches to structure preserving dimensionality reduction," *Fuzzy Systems, IEEE Transactions on*, vol. 10, no. 3, pp. 277–286, Jun 2002.

Soumya Dutta, is a staff scientist at Los Alamos National Laboratory. He received his Ph.D. degree in Computer Graphics and Visualization from the Ohio State University in 2018. His research interests include in-situ data analysis and feature exploration; machine learning and statistical data modeling; uncertainty quantification; time-varying data exploration. Contact him at sdutta@lanl.gov.

Terece L. Turton, is a staff scientist at Los Alamos National Laboratory. Her current research interests include in situ workflows for exascale simulations, scientific visualization, and user evaluation. She received her Ph.D. in Physics from the University of Michigan in 1993. She is a member of the IEEE and the IEEE Computer Society. Contact her at tlturton@lanl.gov.

James P. Ahrens, is a senior scientist at Los Alamos National Laboratory. His research interests include large-scale data analysis and visualizations. He received his Ph.D. in Computer Science in 1996 from the University of Washington. Dr. Ahrens is a member of the IEEE and the IEEE Computer Society. Contact him at ahrens@lanl.gov.