Detection of evolving Lagrangian coherent structures: A multiple object tracking approach

Theodore MacMillan
Department of Aerospace and Mechanical Engineering, University of Notre Dame, Notre Dame, Indiana 46656, USA

Nicholas T. Ouellette
Department of Civil and Environmental Engineering, Stanford University, Stanford, California 94305, USA

David H. Richter[*]
Department of Civil and Environmental Engineering and Earth Sciences, University of Notre Dame, Notre Dame, Indiana 46656, USA

(Received 19 August 2020; accepted 30 November 2020; published 21 December 2020)

Recently, advances in techniques for the detection of Lagrangian coherent structures (LCSs) have driven forward understanding of kinematics and particle transport in a variety of flows. One major shortcoming of these techniques, however, is the lack of an objective procedure for identifying time scales of interest, or an ability to characterize the lives, deaths, or age of coherent structures, especially when relevant flow time scales are larger than the time scales associated with coherence. Here, we address these issues by proposing a multiple object tracking framework frequently used in video analysis to extend the use of existing LCS detection algorithms, allowing coherence time scales to be considered independently of the overall flow evolution time scale and revealing dynamics of the LCSs. Our approach enables the detection of evolving Lagrangian coherent structures, which are created and destroyed throughout the larger time scales of analysis of complex flows.

DOI: 10.1103/PhysRevFluids.5.124401

I. INTRODUCTION

Recent years have seen rapid advances in both the methods for and the uses of the detection of coherent structures in a variety of flows. While instantaneous Eulerian methods like the λ_2 or Okubo-Weiss criteria are useful in certain contexts, such methods have received criticism for their lack of objectivity (frame independence) and poor performance in predicting material transport [1]. To remedy these shortcomings, newer work has instead focused on the identification of Lagrangian coherent structures (LCSs), groups of Lagrangian trajectories which remain dynamically close to each other throughout some characteristic time period. This approach sidesteps both issues with previous Eulerian methods, and has seen success in applications from predicting geophysical transport [2] to search and rescue missions [3].

The basic approach for detecting LCSs is straightforward: Usually, some similarity measure is built between particles in a flow over a fixed time scale τ_C. Next, an algorithm (often borrowed from data science applications) is used to produce clusters drawn from the similarity metric. Hadjighasem *et al.* [4], for example, built a similarity graph between particles using the dynamic interparticle

[*] drichte2@nd.edu

distances over a time interval τ_C. The authors next employ spectral clustering, a well-known data science algorithm [5], to produce groupings of coherent trajectories. Alternative approaches include that of Schlueter-Kuck and Dabiri [6], where interparticle similarity is measured instead using kinematic similarity while ignoring the distance between trajectories, an approach which ensures that similar motions are categorized together rather than just physical proximity. Some approaches avoid interparticle measures altogether and instead focus on Lagrangian measurements of flow quantities. Haller *et al*. [7], for example, sampled the vorticity of particles across a coherence time interval and drew coherent groups from sets of particles with high deviations from the mean flow vorticity during this time interval. There are other methods still, like that of Froyland and Padberg-Gehle [8], which uses another well-known data science algorithm, fuzzy c-means clustering, to extract sets of trajectories from dense regions of the trajectory phase space. These methods and many others are compared by Allshouse and Peacock [9] and Hadjighasem *et al*. [10].

Just as there are many unsolved problems in data clustering generally, certain issues remain with LCS detection algorithms. One major shortcoming of many LCS detection methods, for example, is their lack of temporal resolution. Because LCS detection depends on measuring some Lagrangian metric across a coherence time scale τ_C, there is no clear method for cases where the time scale of analysis, say τ_F, is significantly larger than the time scale of coherence, τ_C. Analyzing coherent structures in turbulent convection is one such example of this predicament, an issue noticed by Schneide *et al*. [11]. As a result, LCS detection methods have primarily been validated with simple, canonical flows like the double gyre or the Bickley jet.

In this paper, we intend to address this shortcoming by utilizing a preexisting algorithm developed for a disparate field; here, we borrow from a group of algorithms developed for use in machine vision known as multiple object trackers. The following basic problem arises in machine vision: Object detection algorithms can identify objects in a single frame of a video, often using techniques like spectral or fuzzy c-means clustering. In consecutive frames of a video, however, objects may enter the view of the camera, move around, or leave altogether. Because a clustering algorithm has no knowledge of the actual objects it is describing, cluster detections at different frames need to be connected to each other if they describe the same actual object. This structure containing many different clusters, all which describe the same object, is referred to as a *track*. The parallels between the problems of video tracking and time-dependent coherent structure detection are clear, and what is missing from LCS methods is the ability to dynamically track the evolution of LCSs in time-dependent flows.

Just as for clustering, there have been a proliferation of algorithms to attack this problem (see Ref. [12] for a review). We further note that this problem is not totally new to fluid dynamics; in fact, many similar algorithms have been developed in the context of Lagrangian particle tracking in flow experiments [13]. We test only one here, an approach which utilizes the Hungarian algorithm to track objects in time. We choose the Hungarian algorithm approach because of its relatively low computational cost, ease of implementation, and the abundance of literature concerning its implementation and operation [14,15].

As we demonstrate, the multiple object tracking (MOT) approach that we outline does not replace current LCS detection techniques; it only extends them to evolving structures. Its performance, therefore, is bound by the fidelity with which LCSs themselves can be detected, and MOT is likewise afflicted with issues affecting clustering algorithms more generally. One such issue which limits the application of LCS analysis in complex flows, for example, is that the number of clusters present in a dataset must often be known *a priori*. While the eigengap heuristic of Hadjighasem *et al*. [4] gives an estimate of the number of clusters, the method seems only suitable for simple flows and well-defined trajectory clusters. Although Schneide *et al*. [11] applied this method for the extraction of turbulent rolls in Rayleigh-Benard convection with some success, the eigengap heuristic as a theoretical concept has been critiqued among data science users for lack of convincing theoretical support and frequent poor performance [16]. Even with these limitations, we show that the application of MOT to track dynamics of LCSs offers rich insights into flow dynamics that are otherwise unreachable via traditional LCS analysis.

In the following section, we detail the steps needed to implement the Hungarian algorithm for the dynamic tracking of LCSs. In Sec. III, we revisit a test case studied by Haller *et al.* [7] and show an added dimension of analysis once evolving LCSs are considered. We then study cases of highly transient LCSs in experimental turbulence. Finally, we conclude in Sec. IV and offer suggestions for future research activity directed at the study of evolving LCSs using MOT.

II. METHODS

A. Clustering

While it is not the focus of our analysis, we first outline the clustering method which will be used for our study. We choose the fuzzy c-means clustering described by Froyland and Padberg-Gehle [8] because of its simple implementation [via the MATLAB fcm( ) function] and its ability to handle large numbers of tracers. We emphasize, however, that the choice of clustering algorithm is somewhat arbitrary, and that the object tracking methodology introduced below can be implemented with other choices of clustering technique.

Beginning with a set of N_p Lagrangian particle trajectories sampled at time intervals $T = (0, 1, \dots, N_t)\Delta t$, we build an array $X \in \mathbb{R}^{N_p \times dN_t}$ where d is the number of spatial dimensions of the data and each row of X contains the concatenated trajectories of each particle over the coherent time interval of interest. The algorithm then randomly initializes k cluster centers C_j in $\mathbb{R}^{dN_t}$ and estimates the "fitness" of these centers (i.e., how close their member trajectories are) by evaluating the objective function

$$\Phi(u, C) = \sum_{j=1}^{k} \sum_{i=1}^{N_p} u_{j,i}^m ||X_i - C_j||^2, \tag{1}$$

where $u_{j,i}$ is the membership value, or the probability that a trajectory i belongs to a cluster j, and $m > 1$ is the fuzziness parameter, controlling by what degree different clusters can overlap. In general, larger values of m will produce clusters with greater membership overlap (or fuzziness), while values of m closer to unity produce clusters which are well delineated.

The initial values of $u_{j,i}$ and C_j can be chosen at random. At each iteration of the algorithm, the positions of the centers are updated by

$$C_j = \frac{\sum_{i=1}^{N_p} u_{j,i}^m X_i}{\sum_{i=1}^{N_p} u_{j,i}^m}, \tag{2}$$

and the membership values are updated by

$$u_{j,i} = \frac{||X_i - C_j||^{-\frac{2}{m-1}}}{\sum_{p=1}^{k} ||X_i - C_p||^{-\frac{2}{m-1}}}. \tag{3}$$

The iterations can be terminated once the objective function given by Eq. (1) falls below a certain value or after a fixed number of iterations are complete. The results of these iterations are a list of membership values $u_{j,i} \in [0, 1]$ for each trajectory i and each cluster j. Each trajectory i is given cluster membership in the cluster j with the largest associated $u_{j,i}$, provided that $u_{j,i} > q$ where q is some fixed threshold. The particles which have no value of $u_{j,i}$ exceeding the threshold q are considered incoherent. Further details of the clustering algorithm can be found in Ref. [8].

Once the threshold q is applied to the membership data, we have divided the N_p particles into k coherent sets and one incoherent set. This step is important as it is generally agreed that coherent flow structures can only be perceived against a background of incoherence [4]. If no q threshold is enforced, every particle becomes a member of some coherent structure. Even with this feature, however, the algorithm still makes no distinctions about over what subset of the N_t points the trajectories exhibit coherence, just that the distance of their position in trajectory phase space is

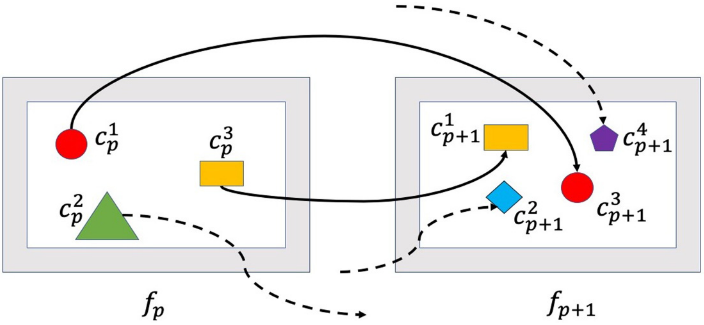

FIG. 1. Object tracking across two intervals, f_p and f_{p+1}. Dotted arrows show objects which are created or destroyed across the two intervals. Solid arrows show objects which persist across the intervals. Note that the numbering of the objects has no correspondence to the actual object being tracked (i.e., the yellow box is object 3 in f_p and object 1 in f_{p+1}).

close to their cluster center. As a result, each particle over the $N_t \Delta t$ time interval can only be a member of a single cluster (neglecting, of course, the fuzzy overlap).

B. Object tracking

With the following steps we extend this clustering approach to the dynamic case, where LCSs are created and destroyed, and particles through their lifetimes may be members of many different coherent structures. From the time steps $T = (0, 1, \ldots, N_t)\Delta t$, we construct a set of Γ overlapping intervals of equal length $F = (f_1, f_2, \ldots, f_\Gamma)$. Each interval consists of α time steps and overlap each other by $\alpha - \beta \geqslant 0$ time steps such that $f_1 = (t_0, t_1, \ldots, t_\alpha)$, $f_2 = (t_\beta, t_{1+\beta}, \ldots, t_{\alpha+\beta})$, and so on. At each interval p, the specified clustering algorithm (fuzzy c-means in our case) is utilized to create a set of k_p unique trajectory clusters, denoted $C_p = (c_p^1, c_p^2, \ldots, c_p^{k_p})$.

Clusters identified at different intervals may describe the same LCS, which itself may grow or shrink as it entrains or ejects fluid particles. The goal, then, is to create a set of n tracks, $R = (r^1, r^2, \ldots, r^n)$, which are themselves sets of clusters from different intervals all which purport to describe the same structure. Figure 1 visually demonstrates this approach.

Because the same structures are composed of approximately the same particles between consecutive intervals, we calculate the Jaccard index between two clusters i and j as $J(i, j) = \frac{\text{number of shared particles}}{\text{number of total unique particles}}$. If i and j describe structures with the same particles, $J(i, j) = 1$, whereas two structures with no similarity will have $J(i, j) = 0$. Two clusters in consecutive intervals which describe the same underlying structure will have a high Jaccard index.

Now, we follow the MOT approach of Weng *et al.* [15] for building the set of tracks. We begin at interval f_1 and initialize k_1 tracks $R = (r^1, r^2, \ldots, r^{k_1})$ with each initially detected cluster $C_1 = (c_1^1, c_1^2, \ldots, c_1^{k_1})$ of the first interval.

In subsequent steps, we attempt to match these clusters with the last cluster associated with each of these tracks. To match a set of clusters C_2 with length k_2 with a set of clusters C_1 with length k_1, we create a $k_2 \times k_1$ Jaccard index matrix, where each entry is the Jaccard index between every cluster pair between C_1 and C_2. Each column of this matrix denotes the Jaccard index of every cluster in C_2 with a single cluster in C_1. The best matching in each column will by definition have the highest Jaccard index, but the optimal pairing is nontrivial as there may be repeats where one cluster in C_2 is the best match for multiple clusters in C_1. This would happen, for instance, in the case where two structures in f_1 merged into a single structure in f_2. The question of finding this optimal pairing solution (a solution where the total Jaccard index is maximized) is called the assignment problem, and is solved using the Hungarian algorithm [17]. The result of this algorithm is a list M of length k_1, where each entry points to the clusters in C_2 which are best fit for each cluster in C_1. If $k_1 > k_2$, there are not enough clusters in C_2 to match each cluster in C_1 so there will be null entries in the list

M. As a guard against poor matching, Weng *et al.* [15] suggested additionally discarding matches below a certain threshold $J_{\min}$, increasing the number of null entries in M.

After initializing at interval $p = 1$, we continue to each subsequent interval and attempt to match the current clusters with the last associated cluster of the current tracks. To allow for misdetections but also to terminate structures once they are gone, we use a tolerance parameter N_D; a track is only terminated if it goes without a match for N_D consecutive intervals. To formalize this process, we define the set R_{last} of length k_t as the set of the last matching cluster of each current track, where a current track is defined as a track which has had a cluster match in the last N_D intervals.

As an example, suppose we have $\Gamma = 4$ intervals, $F = (f_1, f_2, f_3, f_4)$, with cluster sets C_1, C_2, C_3, and C_4. By interval 4, we have $n = 3$ tracks $R = (r^1, r^2, r^3)$, themselves defined by $r_1 = (c_1^1, c_2^3, 0, 0)$, $r_2 = (c_1^2, c_2^1, c_3^2, 0)$, and $r_3 = (c_3^1, c_4^3)$. Note that while r_1 and r_2 in this example were initialized at the first interval, r_3 was not initialized until the third interval. In the case where $N_D = 1$, at the fourth interval we compute $R_{\text{last}} = (c_3^2, c_4^3)$, as the current intervals are r_2 and r_3. If $N_D = 2$, R_{last} would instead become $R_{\text{last}} = (c_2^3, c_3^2, c_4^3)$. The reason for having $N_D > 0$ is to allow for inaccuracies in the clustering method, as real objects may go undetected for a number of intervals before being properly detected again.

At each interval $f_{p>1}$, we use the Hungarian algorithm on the $k_p \times k_t$ matrix formed between the current clusters C_p and the last matching cluster of the current tracks, R_{last}, discarding matches below the threshold $J_{\min}$. After each matching step, we are left with four categories of clusters: those from C_p which have matches, C_p^m; those from C_p which have no matches, C_p^{nm}; those from R_{last} which have matches, R_{last}^m; and those from R_{last} which do not, R_{last}^{nm}. We append the matched clusters in C_p^m to the tracks in R associated with the clusters in R_{last}^m. These are coherent structures which existed in the previous interval and have continued to the current interval. For clusters in R_{last}^{nm}, tracks which have either been destroyed or erroneously missed detection, we append the null character to their associated track in R. Finally, we treat clusters in C_p^{nm} as the potential birth of new clusters and create new tracks in R with each cluster in C_p^{nm} as its initial entry. This approach is summarized in Fig. 2 as well as in Algorithm 1.

Algorithm 1: Detection of evolving LCSs via the Hungarian algorithm.

(1) Break down time steps $T = (0, 1, \ldots, N_t)\Delta t$ into intervals $F = (f_1, f_2, \ldots, f_\Gamma)$ each of length α time steps.

(2) Using any desired clustering algorithm, cluster trajectories at each interval f_p to create Γ sets of k_p clusters, $C_p = (c_p^1, c_p^2, \ldots, c_p^{k_p})$, and create the track list R. The track list will contain lists of clusters which describe the same LCS. Initialize the track list with the initial cluster detections C_1.

(3) Starting at $p = 2$ and for each subsequent interval, create a $k_p \times k_t$ matrix containing the Jaccard index between each cluster detected at interval k_p and the last detected cluster belonging to each track. The last detected cluster of each track is defined as the cluster of a track which has a matched cluster in the past N_D intervals. Use the Hungarian algorithm to find optimal matching between clusters found at frame p and each track and reject matches with a Jaccard index below $J_{\min}$.

(4) Clusters in interval p with no matches are considered possible births and are initialized in the track list R. Tracks with no new cluster match are considered possible deaths and a zero is appended to their cluster membership list.

After iteratively applying this process to each of the Γ intervals, we have n tracks which consist of a collection of clusters from different intervals which describe the same structure. While traditional clustering methods have no temporal resolution, our method has a temporal resolution which is still limited by the size of the intervals α. Thus, α should be the smallest possible value such that $\alpha\Delta t$ still exceeds the coherence time scale τ_C.

A few further notes on particle memberships: over the course of a flow, individual particles may be members of many different tracks. Because the tracks themselves are composed of clusters with generally differing particle memberships (i.e., Jaccard indices between different clusters are less than unity) and may even have null entries for intervals where they were not detected, individual particles may have rather erratic and nonphysical track memberships. For instance, the members

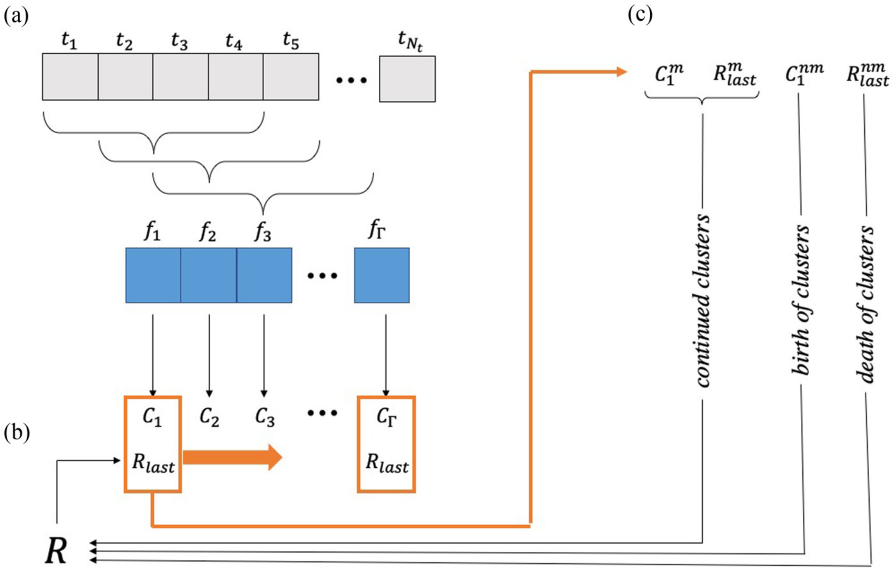
(a)
t_1 t_2 t_3 t_4 t_5 ••• t_{N_t}
f_1 f_2 f_3 ••• f_Γ
(b)
C_1 R_{last} C_2 C_3 ••• C_Γ R_{last}
R
(c)
C_1^m R_{last}^m C_1^{nm} R_{last}^{nm}
continued clusters
birth of clusters
death of clusters

FIG. 2. Overview of our proposed algorithm. (a) N_t time steps of simulation are broken down into Γ intervals of length α and overlap $\alpha - \beta$. In this figure $\alpha = 4$ and $\beta = 1$. Clustering is performed across each of the Γ intervals. (b) R_{last} is created from R as the last associated cluster of all tracks which have had a cluster association in the last N_D time intervals. Clusters in the each interval are associated with clusters in R_{last} using the Hungarian algorithm. (c) The matched clusters, C_p^m, are appended to the tracks associated with the corresponding R_{last}^m. Unmatched clusters from the current interval C_p^{um} are treated as potential births and added to R as new tracks. Unmatched clusters from R_{last}, denoted R_{last}^{um} are treated as potential deaths and the character 0 is appended to their associated track in R.

of an example track $r_1 = (c_5^6, c_6^8, 0, 0, c_9^3)$ will have no clear membership values at intervals 7 and 8, the intervals where no cluster was found with a high enough Jaccard index. To handle these missed detections, we assemble a matrix $P \in \mathbb{R}^{N_p \times N_t \times n}$, where an entry $P_{I,J,K}$ is unity if a particle I is associated with a cluster belonging to track K at time step J and zero otherwise. We can then apply a running average across the time dimension of this matrix and enforce particle membership in tracks when the running average of a particle in some track rises above some threshold. This final step is not necessary for large values of α, but as the temporal resolution is increased (α is decreased) it may become necessary for smoother tracks. This allows for some errors in the cluster detection, as clustering methods not based on strict mathematical rules will not always pick the exact optimal clusters, or even the same clusters given the same data, but we rely on the principle that after many clustering attempts, the algorithm is more likely to identify true clusters than random noise.

Finally, for purposes of time-dependent analysis, special care should be taken with respect to the temporal resolution of our method. Recall that clustering is taken over an interval of α time steps, so the membership of particles is defined over successive *intervals*, not successive time steps. For our analysis, we take the particles to be members of a track for the final time step of an interval over which it belongs to a particular cluster. This arises from the interpretation that the clustering of a particle over α time steps indicates that at any particular time step, it has been coherent over the past α time steps, rather than $\alpha/2$ time steps before and $\alpha/2$ time steps after, or some other combination.

III. RESULTS

We now apply our proposed algorithm to two different test cases. First, we revisit the Agulhas leakage dataset studied by Haller *et al.* [7] and show how an analysis of evolving LCSs offers a

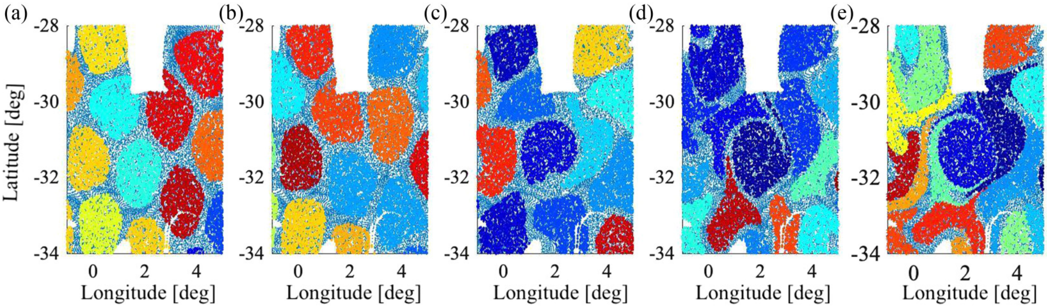

FIG. 3. Example clustering results for (a) $\alpha = 1$, (b) $\alpha = 10$, (c) $\alpha = 50$, (d) $\alpha = 100$, and (e) $\alpha = 200$.

deeper understanding of role of LCSs in the transport of particles in the flow. Then, we analyze an experimental dataset of two-dimensional turbulence to show the strengths of the algorithm in a case where LCSs are extremely transient and are governed by many different time scales.

A. Two-dimensional ocean tracers

We show here the added dimension of analysis available when evolving LCSs are considered. The studied dataset consists of ocean tracers found in the Agulhas leakage in the Southern Ocean between the 90 days [11 November 2006, 9 February 2007], longitudes [−4°, 9°], and latitudes [−35°, −28°]. As the Agulhas current reaches the southernmost point of the African continent, the "leakage" of Indian Ocean water into the Atlantic often exhibits striking coherent structures visible both in ocean circulation models and from remote sensing. This area of the ocean has a particular relevance because of its importance in the global climate system [18] and has been studied in the context of LCS algorithm development because of its distinct vortices, two of which we will analyze. Tracers are integrated by a fourth-order Runge-Kutta method through satellite-obtained velocity fields and their positions are sampled at 600 evenly spaced time intervals. A complete description and a link to the data used in this study are found in Ref. [7].

It is, in principle, difficult to systematically choose a value of the interval length α with which to analyze a flow. In traditional LCS analysis, this problem is no less present, as α is chosen by default as the duration of the flow being analyzed. Furthermore, the effect of α on the transient LCSs detected will vary with the method of clustering which is used. Fuzzy clustering with a Euclidean distance metric as its objective function, for example, will cluster together trajectories which are spherical in their $d \times \alpha$ dimension trajectory space. With this distance metric, choosing $\alpha = 1$ will produce clusters that are approximately spherical in the two-dimensional trajectory space, or circular clusters. More fundamentally, the selection of a small α cannot capture any of the coherence information that a LCS detection technique ought to. An overly large choice of α, however, will lead to a loss of coherence as additional noncoherent trajectory points are used in the clustering. Figure 3 shows the clusters obtained for values of increasing α and serves as a visual selection criterion.

In general, α should be selected as small as possible without falling below the coherence time scale of the flow. This ensures that our MOT method can uncover the transient dynamics of LCSs while still discovering the relevant coherent structures. To illustrate this point and for the sake of comparison, once the tracers are sufficiently developed, we apply fuzzy c-means clustering over the entire time interval $T = [30, 75]$ days using clustering parameters $m = 2$, $q = 0.1$, $k = 30$, and $\alpha = N_t$ (α is the entire duration of the flow). This is similar to what a typical LCS analysis might consist of. For simplicity, we only consider the two central counter-rotating vortices of the dataset. Figures 4(a)–4(d) show the clustering results for these two vortices over this time interval. As evident from the figure, this static clustering will only group particles that are coherent across the total time interval. The consequence of this is that yellow particles can be found inside the red vortex since the method is not sensitive to particles which are members of different coherent

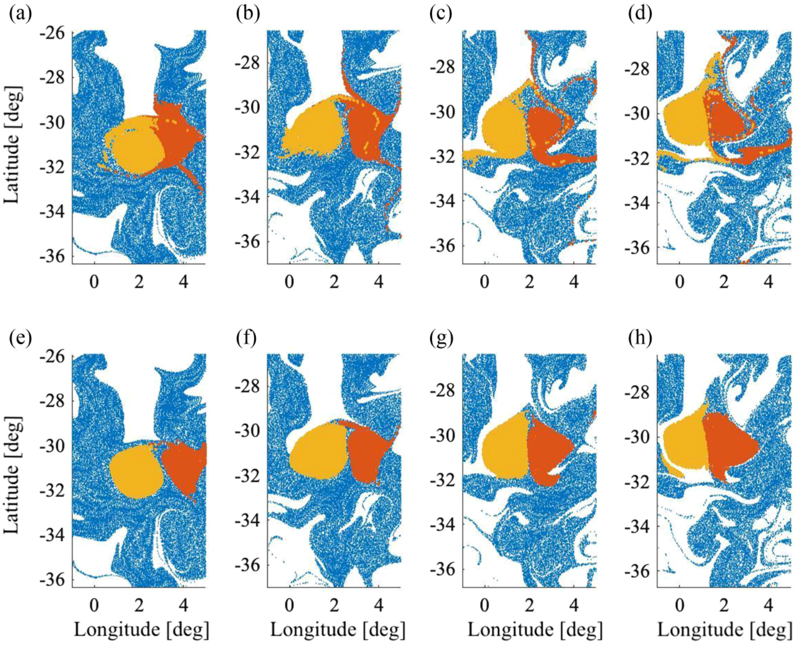

FIG. 4. (a–d) Static clustering results across the interval $T = [30, 75]$ days. Moving right, the same static clustering results are displayed at day 30, day 45, day 60, and day 75. (e–h) Dynamic clustering across the same time period and shown at the same days using MOT. In both cases, the left vortex is displayed in yellow and the right vortex is displayed in red.

structures during their lifetimes. In contrast, we show our object tracking results in Figs. 4(e)–4(h): using a minimum coherence interval $\alpha = 100$ (15 days), our method is sensitive to the changing shape of the two different vortices.

We can demonstrate, for example, that the sizes of these two vortices are changing as particles dynamically enter and exit the structures. Figure 5 shows the number of particles contained in each cluster as time evolves, confirming that the right vortex is siphoning membership from the left vortex; this is qualitatively the mixing seen in the static Figs. 4(a)–4(d). Since it is evident that the left vortex is shedding particles while the right vortex is entraining them, we can further analyze this behavior by looking at the tracks' respective dynamic particle membership. Figure 6 shows the clusters at day 68 with their current particle members colored according to the time since they were first members in the vortex. Since the left vortex is mostly decreasing in particle count, all the particles which are currently members of the vortex have been there for the same time (i.e., since the birth of the cluster; roughly 35 days). In contrast, the right vortex exhibits several distinct layers of particles. Each layer has a distinct time signature which is sharply delineated from surrounding layers. These dynamics are effectively hidden from a clustering analysis that only looks at coherence across the total time interval, and yields insight into the dynamics of the coherent structure itself.

To further emphasize this point, in Fig. 7 we visualize the particles which cross over from the left vortex into the right vortex over the time interval considered. It is qualitatively clear that the particles which switch from the left to the right vortex represent the principle differences between static clustering results and dynamic clustering results (see Fig. 4).

B. Two-dimensional experimental turbulence

Our next subject of analysis is a dataset of experimental two-dimensional turbulence obtained via particle tracking velocimetry. The time interval we analyze consists of tracers advected over

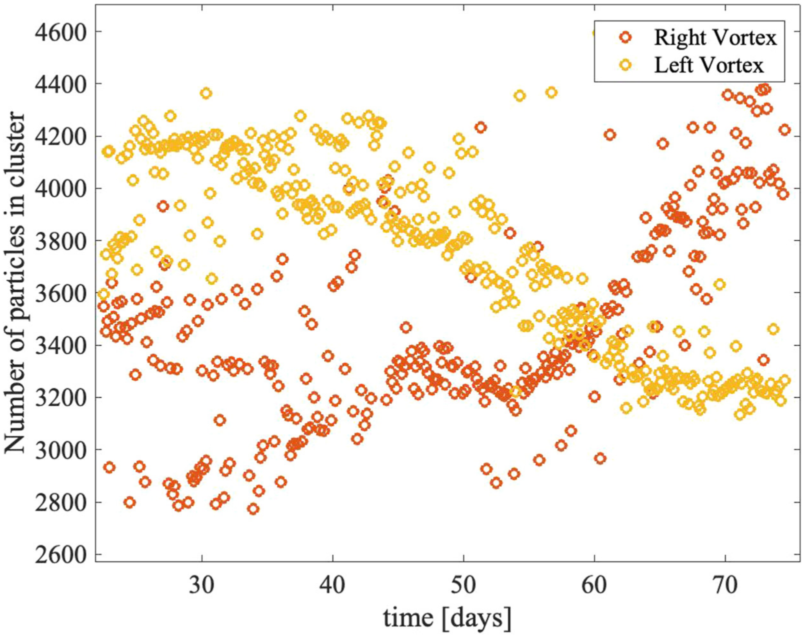

FIG. 5. Plot of cluster membership against time for the two central vortices.

the time interval $T = [0, 34]$ s, with a characteristic eddy turnover time of 2.9 s. The methods and postprocessing applied to the original PTV are described fully in Ref. [19]. Tracers are integrated across the available velocity fields using Euler's method and sampled at $N_t = 200$ evenly spaced points within the considered time frame.

In this application, we demonstrate the ability of our proposed algorithm to handle large numbers of transient LCSs, since this is a more complex dataset than the ocean tracer set in the previous section. In particular, this example provides a case where the time scale of analysis is much larger than the time scale of coherence, or $\tau_F \gg \tau_C$. In this case, our known $\tau_C \approx 2.9$ s sets a lower limit on the scale we want to resolve and track through a time τ_F; i.e., one may want to describe quantities which evolve over the course of the entire experiment, but clustering over this entire time frame

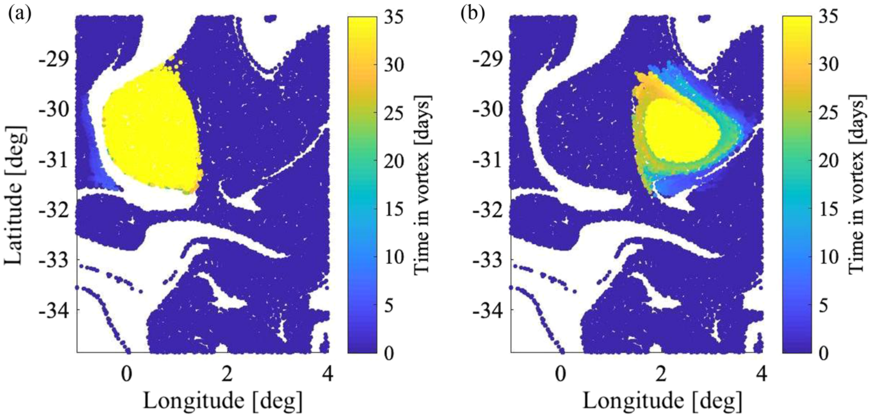

FIG. 6. Snapshot of tracer particles on day 67. (a) Left central vortex. (b) Right central vortex. Current particle members are colored by how long they have been inside the displayed structure.

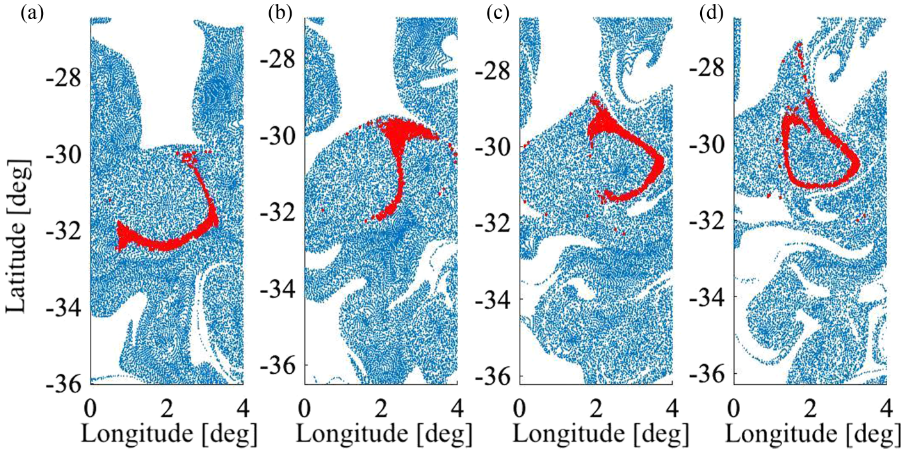

FIG. 7. Visualization of tracers (red) which cross over from the left vortex to the right vortex. Times displayed are for (a) day 30, (b) day 45, (c) day 60, and (d) day 75.

would be meaningless as particles will only remain in groups for a small fraction of the total time frame.

At a first glance, we can visually inspect the coherence of this flow by drawing unbroken trajectories over varying numbers of time steps. As with our last example, this serves to qualitatively guide a selection of α. For our analysis we pick $\alpha = 30$, creating intervals which last 5.1 s. Figure 8 displays the MOT results using these values, along with fuzzy c-means clustering parameters $k = 50$, $m = 1.5$, and $q = 0.9$.

As noted in the previous example, clustering methods often show sensitivity to changes in various parameters. Schlueter-Kuck and Dabiri [20], for example, showed that spectral clustering only exhibits an accurate spectral gap and correct cluster count when the interparticle similarity matrix is properly sparsified. Fuzzy c-means clustering also has several tunable parameters which may influence its clustering results. This raises a concern for noncanonical flows where the *a priori* number of structures cannot be guessed by visual inspection. By comparing Figs. 8(a)–8(c) with Figs. 8(d)–8(f) we can see that many coherent flow structures are captured, but our MOT approach cannot outperform the clustering method it uses to identify LCSs in the first place; i.e., the tracking algorithm is fundamentally limited by the accuracy of the underlying clustering method. The clustering results at each of our intervals are very similar to those obtained by Hadjighasem *et al.* [10] using the same clustering method, but the objective validity of LCSs detected are hard to quantify beyond the subjective opinion of an observer.

There are, however, some methods for quantitatively assessing the accuracy of LCS detection methods. Beron-Vera *et al.* [2], for example, calculated the mean pairwise distance of particles inside detected LCSs. For our clustering method, however, this is to be expected, since by definition particles in the same LCS have shown a higher degree of spatial coherence than the outside flow. Therefore, we instead take an approach utilized by Fang *et al.* [21], where coherent structures are checked by using a quantity unrelated to their definition. Visual inspection of our object tracking results (Fig. 8) shows that many of the detected structures appear to be rotational in nature, so we employ the instantaneous vorticity deviation (IVD) metric of Haller *et al.* [7], defined as

$$\mathrm{IVD}(x,t) = |\omega(x,t) - \bar{\omega}(t)|, \tag{4}$$

where ω is vorticity and the overbar denotes an average across all particles. Because the turbulent flow considered has distinct vortices, many of which are created and destroyed as the flow progresses in time, we can measure the IVD experienced by a particle in a particular evolving LCS before, during, and after it is detected as part of this respective structure. This is visualized in Fig. 9. Here,

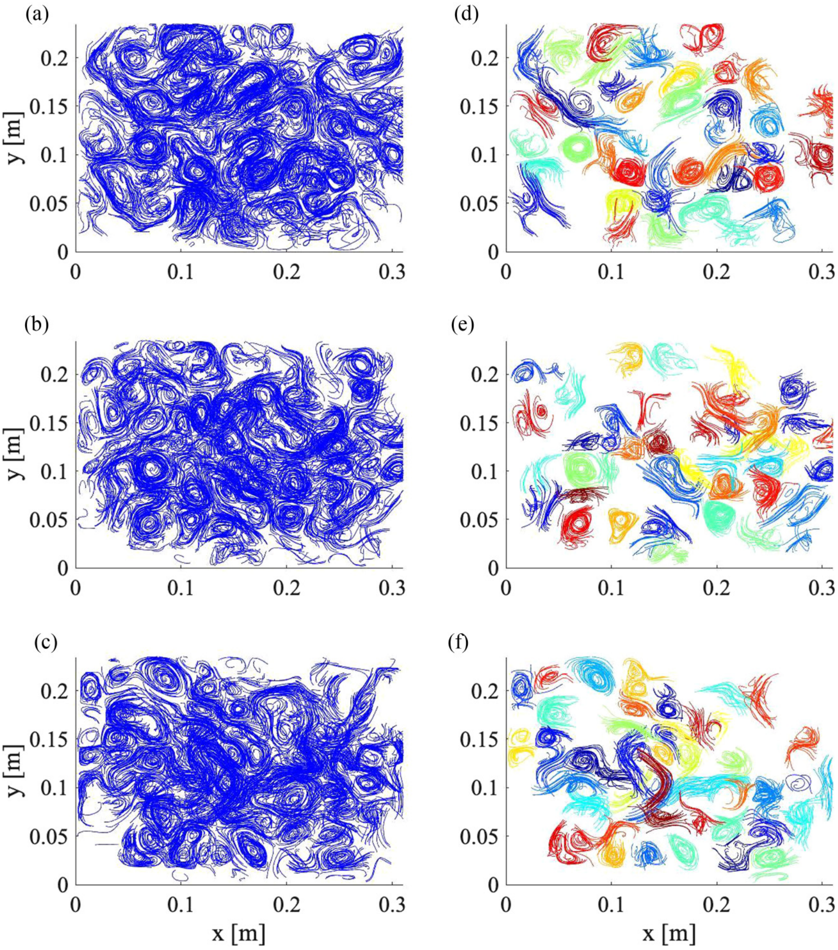

FIG. 8. Tracer trajectories drawn over the interval duration $\alpha = 30$ time steps. (a–c) All tracers in the flow at respective intervals f_1, f_{51}, and f_{101}. (d–f) Tracers colored by their track membership at the same intervals, excluding tracers without any tracks.

$t = 0$ refers to the time when a cluster is first identified, and the transparent curves refer to the average IVD of all particles which are a member of it, before and after its initiation. As one may expect, particles experience the highest IVD in the time shortly after they are detected as being part of a coherent structure. Figure 9 demonstrates that the object tracking algorithm outlined in this work can identify the time evolution of kinematic flow features associated with the LCSs, and shows that a characteristic lifetime of the coherent motions exceeds roughly 20 s (i.e., when the mean IVD decreases to predetection levels). In the case of the two-dimensional turbulence dataset, the peak of IVD shortly after cluster identification and its eventual decay provides insight into the time response of these vortical regions. This is noteworthy especially since the clustering algorithm has no knowledge of the flow vorticity.

As a final result, we note that the outlined MOT approach can also track the merging and splitting of coherent structures. Figures 10 and 11 show merging and splitting events, respectively, and offer insights into how the Hungarian algorithm may handle these cases. In Fig. 10, two independent structures are detected for several intervals and eventually come together into a single structure. In an intermediate step, however, a third structure is detected between the original two and its identity

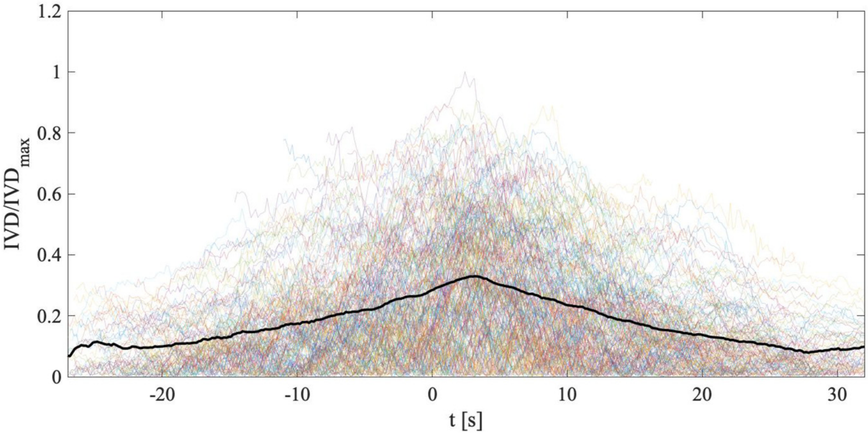

FIG. 9. Plot of IVD against time for different clusters. Each line shows the mean IVD of particles that are members of particular evolving clusters, and $t = 0$ is when the object is formally detected. As expected, the high IVD for the particles in the tracks is reached around the same time they are detected as a track. The mean of all of tracks is shown in black.

is maintained as the eventual merged structure. In Fig. 11, a single structure splits into two smaller structures. As the Hungarian algorithm assigns continuity based on the Jaccard index, the larger of the two child structures will maintain the identity of its parent and the smaller will become a new track. The combined information contained in Figs. 9–11 thus highlights the prospects of using this algorithm for understanding the life cycles of these coherent structures.

IV. CONCLUSION

In this paper, we have presented a method for extending traditional LCS analysis to the detection and study of time-dependent, evolving LCSs. In particular, by dividing a dataset into many unique

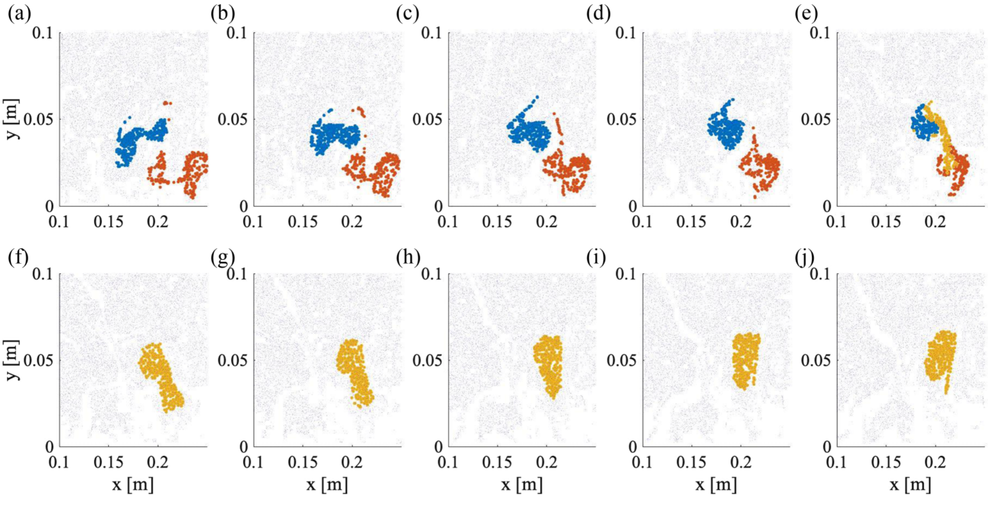

FIG. 10. Merging event observed in the turbulence data set. As the two initial structures (red and blue) approach one another, a third structure is detected and assimilates the two smaller structures. Time advances from (a) to (j).

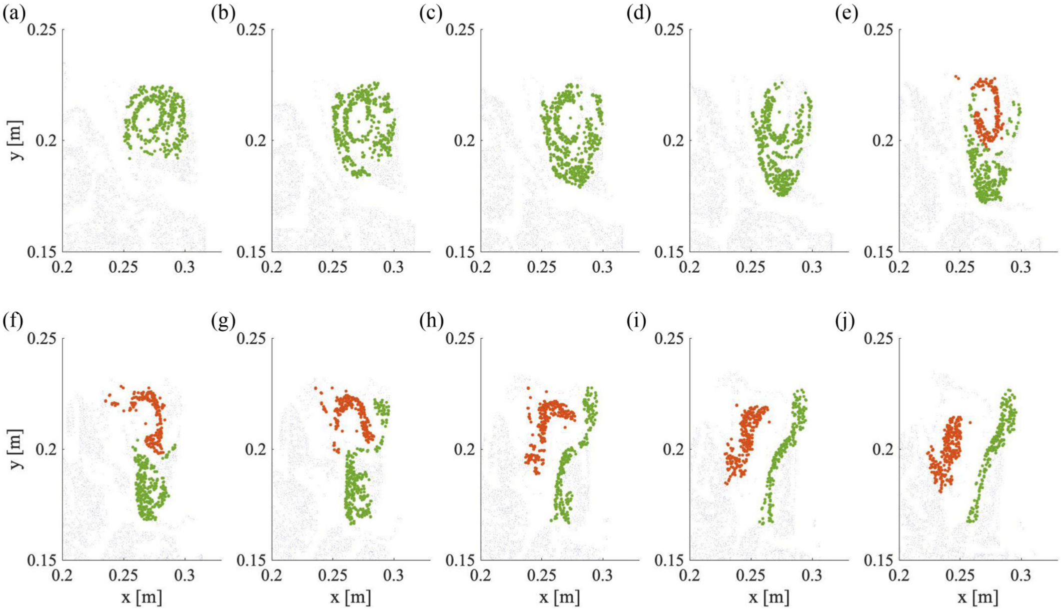

FIG. 11. Splitting event observed in the turbulence data set. The initial structure (green) breaks into two smaller structures (red and green). The larger of the two maintains the identity of the initial structure. Time advances from (a) to (j).

coherence intervals, we allow existing LCS detection methods to deploy on unsteady flows with coherent structures of varying sizes and time scales. We explore the uses of our algorithm together with fuzzy c-means clustering, but we stress that other LCS approaches are equally or more suitable for different problems. For example, we anticipate that detection approaches like Lagrangian-averaged vorticity deviation would be very successful with our MOT method due to its mathematically well-defined clusters, which would not likely deviate significantly between intervals.

While we demonstrate that the evolving coherent structures which we detect exhibit more coherence than the mean flow, it is still unclear how much parametric changes in both the clustering method and the MOT method will alter statistics associated with the transient clusters. Along these lines, flows which have less verifiable coherent structures pose a difficulty to our proposed MOT method. Three-dimensional turbulence, for example, attracts fewer LCS analyses because it is less straightforward to verify the ground truth of coherence. Traditional LCS analysis, of course, shares this weakness, but it is amplified in our algorithm because subsequent intervals may yield incorrect or conflicting cluster identifications. To improve our MOT algorithm, LCS methods which have fewer parameters and more verifiable results are needed. We further note that the extraction of temporal statistics is not straightforward because coherence even in the tracks is not defined at time steps, but rather time intervals. These are not, of course, issues just experienced within LCS detection, but problems in object detection more generally, and we leave it to future work to quantify the importance of these parametric changes to the ground truth of evolving LCSs.

ACKNOWLEDGMENTS

T.M. and D.R. acknowledge Grant No. G00003613-ArmyW911NF-17-0366 from the U.S. Army Research Office and Grant No. N00014-16-1-2472 from the Office of Naval Research. Computational resources were provided by the Center for Research Computing (CRC) at the University of Notre Dame.

[1] G. Haller, Lagrangian coherent structures, Annu. Rev. Fluid Mech. **47**, 127 (2015).

[2] F. J. Beron-Vera, A. Hadjighasem, Q. Xia, M. J. Olascoaga, and G. Haller, Coherent Lagrangian swirls among submesoscale motions, Proc. Natl. Acad. Sci. **116**, 18251 (2018).

[3] M. Serra, P. Sathe, I. Rypina, A. Kirincich, S. D. Ross, P. Lermusiaux, A. Allen, T. Peacock, and G. Haller, Search and rescue at sea aided by hidden flow structures, Nat. Commun. **11**, 2525 (2020).

[4] A. Hadjighasem, D. Karrasch, H. Teramoto, and G. Haller, Spectral-clustering approach to Lagrangian vortex detection, Phys. Rev. E **93**, 063107 (2016).

[5] U. von Luxburg, A tutorial on spectral clustering, Stat. Comput. **17**, 395 (2007).

[6] K. L. Schlueter-Kuck and J. O. Dabiri, Identification of individual coherent sets associated with flow trajectories using coherent structure coloring, Chaos **27**, 091101 (2017).

[7] G. Haller, A. Hadjighasem, M. Farazmand, and F. Huhn, Defining coherent vortices objectively from the vorticity, J. Fluid Mech. **795**, 136 (2016).

[8] G. Froyland and K. Padberg-Gehle, A rough-and-ready cluster-based approach for extracting finite-time coherent sets from sparse and incomplete trajectory data, Chaos **25**, 087406 (2015).

[9] M. R. Allshouse and T. Peacock, Lagrangian based methods for coherent structure detection, Chaos **25**, 097617 (2015).

[10] A. Hadjighasem, M. Farazmand, D. Blazevski, G. Froyland, and G. Haller, A critical comparison of Lagrangian methods for coherent structure detection, Chaos **27**, 53104 (2017).

[11] C. Schneide, A. Pandey, K. Padberg-Gehle, and J. Schumacher, Probing turbulent superstructures in Rayleigh-Benard convection by Lagrangian trajectory clusters, Phys. Rev. Fluids **3**, 113501 (2018).

[12] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, X. Zhao, and T.-K. Kim, Multiple object tracking: A literature review, arXiv:1409.7618.

[13] N. T. Ouellette, H. Xu, and E. Bodenschatz, A quantitative study of three-dimensional Lagrangian particle tracking algorithms, Exp. Fluids **40**, 301 (2006).

[14] B. Sahbani and W. Adiprawita, Kalman filter and iterative-Hungarian algorithm implementation for low complexity point tracking as part of fast multiple object tracking system, *2016 6th International Conference on System Engineering and Technology (ICSET)* (IEEE, Piscataway, NJ, 2016), pp. 109–115.

[15] X. Weng, J. Wang, D. Held, and K. Kitani, 3D multi-object tracking: A baseline and new evaluation metrics, arXiv:1907.03961.

[16] L. Zelnik-Manor and P. Perona, Self-tuning spectral clustering, Adv. Neural Inf. Process. Syst. **17**, 1601 (2005).

[17] H. W. Kuhn, The Hungarian method for the assignment problem, Nav. Res. Logistics Q. **2**, 83 (1955).

[18] L. M. Beal, W. P. De Ruijter, A. Biastoch, R. Zahn, M. Cronin, J. Hermes, J. Lutjeharms, G. Quartly, T. Tozuka, S. Baker-Yeboah, T. Bornman, P. Cipollini, H. Dijkstra, I. Hall, W. Park, F. Peeters, P. Penven, H. Ridderinkhof, and J. Zinke, On the role of the Agulhas system in ocean circulation and climate, Nature **472**, 429 (2011).

[19] D. H. Kelley and N. T. Ouellette, Onset of three-dimensionality in electromagnetically driven thin-layer flows, Phys. Fluids **23**, 45103 (2011).

[20] K. L. Schlueter-Kuck and J. O. Dabiri, Coherent structure colouring: Identification of coherent structures from sparse data using graph theory, J. Fluid Mech. **811**, 468 (2019).

[21] L. Fang, S. Balasuriya, and N. T. Ouellette, Local linearity, coherent structures, and scale-to-scale coupling in turbulent flow, Phys. Rev. Fluids **4**, 014501 (2019).