DOI: 10.1109/CJECE.2019.2917394

Evolution Surfaces for Spatiotemporal Visualization of Vortex Features

Simon Ferrari, *Student Member*, *IEEE*, Yaoping Hu, *Member*, *IEEE*, and Robert J. Martinuzzi

***Abstract*—Turbulent fluid flow data is often 4-dimensional (4D), spatially and temporally complex, and requires specific techniques for visualization. Common visualization techniques neglect the temporal aspect of the data, limiting the ability to convey feature motion or offering the user a complicated visualization. To remedy this, we present an approach – evolution surfaces – focused on the spatiotemporal rendering of user-selected flow features (i.e., vortices). By abstracting the spatial representation of these features, the approach renders their spatiotemporal behavior with reduced visual complexity. The behavior of vortex features are presented as surfaces, with textures indicating properties of motion and evolution events (e.g., bifurcation and amalgamation) represented by the surface topology. We evaluated the approach on two datasets generated from empirical measurement and computational simulation (*Re* = 28000 and *Re* = 1200 respectively). Our approach's focus on handling evolution events makes it capable of visualizing higher Reynolds number (*Re*) flows than other surface-based techniques. This approach has been assessed by fluid dynamicists to assert the validity for flow analysis. Evolution surfaces offer a compact visualization of spatiotemporal vortex behaviors, opening potential avenues for exploration and analysis of fluid flows.**

***Index Terms*— feature tracking, flow analysis, spatiotemporal visualization, vortex extraction**

I. Introduction

Visualization of turbulent fluid flow data requires the ability to convey complex spatiotemporal properties to the user. Outside of flow visualization, temporal relationships are commonly explored by the addition of a time dimension, enabling temporal pattern recognition [1, 2]. However, for the 3-dimensional (3D) features of interest within flow data different tactics for illustrating these patterns must be used. While the intuitive solution is to separate spatiotemporal change into a series of images, this can end up limiting the accuracy of the user's analysis [3]. Existing spatiotemporal visualization techniques for volumetric flow features struggle with visual complexity, limiting the temporal resolution of the image and the accuracy of the user's analysis [4].

In this paper we present a novel approach to visualizing the spatiotemporal evolution of features with reduced visual complexity. The "evolution surface" approach abstracts flow features into a set of representative ridges, and renders the temporal behavior of the feature as a contiguous surface. Evolution events such as bifurcations [5] are modelled as changes in the surface topology, and properties of motion (e.g., changes in velocity) are presented through textures. These surfaces can identify any potential evolution event, and do so with less computational overhead than other spatiotemporal surface techniques [6, 7]. This compact visualization permits rendering of feature behavior over long time periods, as well as presenting multiple features simultaneously for comparing behavior patterns between experimental conditions.

This paper focuses on the extraction and tracking of vortices due to their importance to engineering design work. Vortices contain low pressure, high energy, and are responsible for mixing and a variety of forces on objects such as lift or drag [8, 9]. Aerodynamic design often involves the study of vortices in the wake of an object to manipulate these forces (e.g., controlling thrown debris) [10, 11]. As vortices often take the form of tube-like structures rotating about a central core, they are ideal for abstraction into ridges. In this paper we've applied evolution surfaces to two high Reynolds number (*Re*)[1] flows of interest from experimental and simulation data (*Re* = 28000 and *Re* = 1200 respectively).

II. Related Work

There has been a variety of work conducted in characterizing temporal visualization, and how it can assist the user in learning or analysis tasks [1, 2]. Most of the techniques for spatiotemporal visualization rely on the extension of an axis to represent time, although this is only suitable for 1D or 2D data. Two surveys focused on the state of the art in flow visualization are Post et al. [13] and Brambilla et al. [14]. We have identified the main spatiotemporal flow visualization techniques and applied some to a simple flow feature within a modified Arnold-Beltrami-Childress flow field [15] in Fig. 1.

The most straightforward and widely used representation of

Manuscript submitted June 22, 2018. This work was supported in part by the Natural Science and Engineering Research Council of Canada under Grants: RGPIN-2015-06601 to Y. Hu and STPGP 478870 to Y. Hu and R. Martinuzzi.

S. Ferrari and Y. Hu are with the Department of Electrical and Computer Engineering at the University of Calgary, in Calgary, Alberta, CANADA. (email: swferrar@ucalgary.ca, huy@ucalgary.ca).

R. J. Martinuzzi is with the Department of Mechanical and Manufacturing Engineering at the University of Calgary, in Calgary, Alberta, CANADA. (email: rmartinu@ucalgary.ca).

[1] The Reynolds number [12] represents the relationship between velocity and viscosity of a fluid, and is often used to describe the turbulence of a flow.

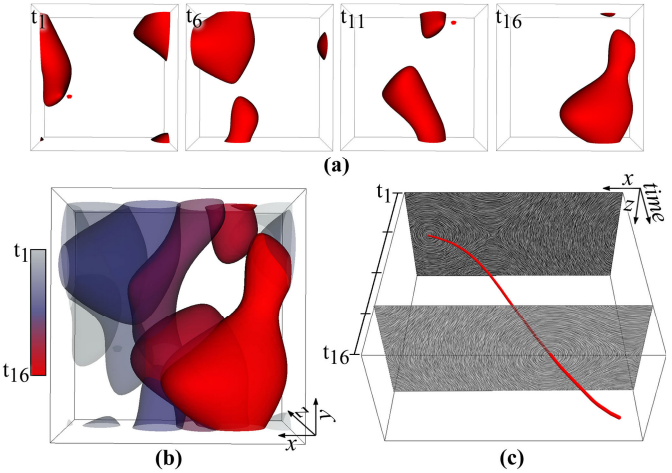

Fig. 1. Temporal evolution of a flow feature over 16 time instants visualized using: (a) still images, (b) chronovolume, and (c) STC.

spatiotemporal change in a flow is through sequential images of time instants. When spatially separated these time instants form a time series as in Fig. 1(a); when temporally separated they become an animation. While convenient due to their use in print media, still images struggle to convey properties of motion (e.g., velocity). Space constraints often restrict the number of available images, reducing the user's ability to comprehend the transitional states of complex behaviors. Animation is the most intuitive means of presenting change over time, however users can only focus on a limited number of features at a time, limiting its use as an analysis tool [3, 16].

To allow the user to better intuit the transitional states of temporal data, chronovolumes [17] collapse multiple time instants into the same volume, as shown in Fig. 1(b). By removing the spatial separation present in still images, properties of motion are more obvious, and by being static, features are less arduous to track [18]. Unfortunately, feature overlap causes chronovolumes to be prone to high visual complexity and occlusion with increasing temporal resolution [19]. Joshi and Rheingans [20] explored alternative methods for abstracting the transitional states of features through lines based on illustrations and comics. (i.e., speed lines). While this method reduces visual complexity, its capabilities of expressing complex spatiotemporal evolution events were limited.

Space-time cubes (STC) [21, 22] Offer a high temporal resolution visualization with low visual complexity. STC track the movement of points of interest along a 2D plane, with the time instants distributed such that one axis depicts time, as in Fig. 1(c). STC is useful for aiding the user to understand spatiotemporal interactions and the timing of events. STC has been used in flow visualization to track critical points [23] and evolution of more complex contour-features [24]. These changes are presented as lines or surfaces, but are limited to visualizing 2D features.

Where still images and chronovolumes suffer is due to the visual complexity of achieving a high temporal resolution due to the volumetric nature of the features of interest. The rendering of spatiotemporal properties within the flow volume with lower-dimensional primitives (similar to STC) can reduce

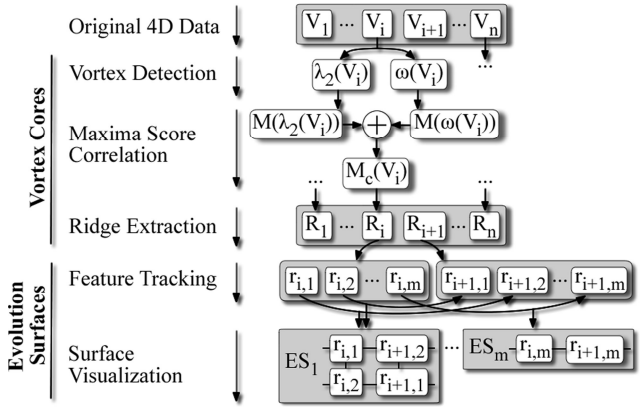

Fig. 2. The processing pipeline of the evolution surface approach.

this visual complexity. When used en masse, pathlines or path surfaces tracking the motion of particles which intersect the features of interest can provide insight into the temporal behavior of the flow [7, 25], though pathlines on the velocity field do not directly track feature movement.

Feature flow fields (FFF) [23] offer a means to track features through advected pathlines or path surfaces. FFFs are vector fields derived from the original velocity data where streamlines seeded at critical points follow the path taken by those points over time. FFF's are most commonly used to produce an STC of 2D features, but when used in conjunction with parallel vectors (PV) they can extract line-features. In extending the concept to 4D, they have been used to visualize the behavior of vortex cores as stream surfaces [6]. These surfaces are low visual complexity at a continuous temporal resolution, however the way in which they are generated presents some caveats. First, the computation time for a 4D FFF field is considerable due to the high-order derivatives required. Second, the PV lines and advection have limited capacity for detecting evolution events which do not occur at the data's boundaries. As such, this method has been reported only on low Reynolds number flows with relatively simple vortex shedding patterns (Re = 300) [6]. Nevertheless, there is a lack of techniques capable of visualizing vortex evolutions without compromising temporal resolution or increasing visual complexity.

III. Approach

To produce a visualization of feature behavior with low visual complexity that can handle the more complex feature behaviors present at higher Reynolds numbers, we developed the evolution surface approach. Evolution surfaces operate by tracking ridge lines abstracting volumetric features, and rendering their path as a surface. In the following we present our approach to 3D vortex core extraction based on image processing techniques, followed by our tracking and rendering algorithms. For this approach, the features of interest are coherent turbulent structures [26]. This implies that the features are spatially distinct to enable extraction and have temporal coherencies over their lifespan for tracking. These features are of great importance to engineering design work [8-11]. The complete processing pipeline is given in Fig. 2, which expands on preliminary work with these surfaces [27].

A. Vortex Detection

Despite the prominence and importance of vortices to flow dynamics, there is no universal definition for how to mathematically detect them [8, 9, 28, 29]. However, it is commonly known that vortices forming coherent turbulent structures are detectable by using methods such as the vorticity $\vec{\omega}$ [30], the λ_2 criterion [31], and/or the Q criterion [32]. These methods are widely used for Eulerian vortex detection. Although both the λ_2 and Q criteria detect similar vortices in most cases, the λ_2 criterion appears to detect vortices in some cases that the Q criterion could fail [31]. Thus, to improve the robustness of our feature ridge lines we employed the vorticity $\vec{\omega}$ and the λ_2 criterion. These vortex detectors are both applied on a given velocity field V_i at time t_i, where the data consists of a series of velocity vectors $\vec{v}$ on each point of a rectilinear 3D grid (Cartesian components x, y, and z). The ridge lines based on vortices detected through these two methods are compared later with those via the Q criterion (Section IV).

The vorticity $\vec{\omega}$ is a vector which describes the direction and magnitude of rotation of a velocity field [30]:

$$\vec{\omega} = \nabla \times \vec{v}\,, \tag{1}$$

where ∇ is the del operator. A larger magnitude $\|\vec{\omega}\|$ indicates stronger rotation. Due to attenuation (i.e., the action of diffusive and mixing processes), the intensity of vorticity in a wake can vary greatly depending on proximity to the turbulence-inducing obstruction, reducing the effectiveness of this method to identify and track vortices.

Rotational motion of the flow results in local pressure minima, which can signify the locations of vortices. The λ_2 criterion [31] utilizes the rotation Ω and strain S of the velocity gradient tensor $\nabla\vec{v}$ as follows:

$$\Omega = 0.5(\nabla\vec{v} - (\nabla\vec{v})^t)\,, \tag{2}$$

$$S = 0.5(\nabla\vec{v} + (\nabla\vec{v})^t)\,. \tag{3}$$

The tensor $\Omega^2 + S^2$ represents the part of the pressure Hessian related to local pressure caused by vortex motion. When the three real eigenvalues of $\Omega^2 + S^2$ are ordered as $\lambda_1 \le \lambda_2 \le \lambda_3$, a point may be part of a vortex if $\lambda_2 < 0.0$ and $\lambda_3 > 0.0$. Since it can better cope with various intensities of vortex features than vorticity, the λ_2 criterion is widely used in flow analysis.

B. Maxima Score Correlation

The maxima score is a normalized ridge detector that produces a value directly related to the configuration of scalar intensities between a point and its neighbors [33]. The computation time of this detector is comparable to methods employing Hessian eigenvalues, a widely accepted detector for ridge extraction [34 – 39]. As a benefit over Hessian approaches, the value produced by the maxima score shares the same configuration-based meaning between datasets. This allows the maxima score to be used without specific knowledge of the dataset and simplifies correlation between domains. When applied to vortex detection, such local extrema have served to visualize vortex core lines as ridges [30, 38, 40 – 42]. The local extrema have an advantage of avoiding the use of high-order derivatives for extracting core lines [43].

Applied to a scalar field F, the maxima score produces a normalized field of scalar values $1.0 \le M(F) \le -1.0$, where the extremes indicate local minima and maxima, respectively. The scalar value $M(q_0)$ represents the similarity of the quantity q_0 to a local maximum or minimum. $M(q_0)$ is obtained by comparing the quantity q_0 (at the target point) to each of its n neighbor quantities q_i for $i = \{1, \ldots, n\}$ within a defined volume (the "local neighborhood") as follows:

$$M(q_0) = \frac{\sum_{i=1}^{n} m(q_0, q_i) \times G(q_0, q_i)}{\sum_{i=1}^{n} G(q_0, q_i)}, \tag{4}$$

$$m(q_0, q_i) = \begin{cases} 1, & q_0 - q_i > T_M \\ -1, & q_0 - q_i < -T_M \\ 0, & else \end{cases}, \tag{5}$$

where T_M is a positive threshold to determine if the difference between the quantity of q_0 and its neighbors is significant. This threshold is dynamically set to approximately 1.0% of the scalar intensity of the feature under study, which reduces the effect of small fluctuations while preserving feature detail [27]. Differing from prior work [33], we use a modulator $G(q_0, q_i)$ to emphasize the relative intensity of nearby points to the target point. The value of $G(q_0, q_i)$ is the relative position of the target and neighbor point within a 3D Gaussian filter ($\sigma = 1.5$). For a rectilinear grid with minor variances of dx, dy, and dz, a 5 x 5 x 5 cube centered on the target point serves as the volume.

Prior work indicates that combining the maxima scores of the vorticity and λ_2 criterion increases robustness of vortex detection over either individual domain [33]. Given $M(\|\vec{\omega}\|)$ and $M(\lambda_2)$ as the maxima scores of the vorticity and λ_2 criterion, respectively, the correlated maxima score field at t_i is:

$$M_c(V_i) = w_\omega M(\|\vec{\omega}\|) + w_{\lambda_2} M(\lambda_2)\,, \tag{6}$$

where w_ω is the weight of $M(\|\vec{\omega}\|)$ with a value of 0.442 for detecting local maxima and w_{λ_2} represents the weight of $M(\lambda_2)$ with a value of -0.558 for detecting local minima. These weights were derived from a survey conducted amongst fluid dynamicists for relative pertinences of the vorticity and λ_2 criterion to study vortices [33]. Importantly, $M(\|\vec{\omega}\|)$ is an objective measure (i.e., invariant to translation and rotation) and $M(\lambda_2)$ is Galilean invariant (i.e., unchanged by translation) [29]. As a result, the correlated domain $M_c(V_i)$ is Galilean invariant to enable tracking extracted vortices through the flow field.

C. Ridge Extraction

To facilitate the reduced visual complexity of the evolution surfaces, the 3D vortex features are abstracted into 1D ridges. Applied to $M_c(V_i)$, these ridges describe the shape of the coherent vortex features. Algorithm 1 describes our curve-tracing algorithm, which uses hysteresis thresholds commonly found in image processing [44]. Ridges are grown iteratively along the paths of highest intensity as identified by the maxima score. By growing ridges iteratively, the interference between nearby features is reduced.

This iterative ridge growing uses current thresholds (*CurrSeed* and *CurrRidge*) and target thresholds (*SeedThresh* and *RidgeThresh*). The current thresholds are initially set restrictively, and reduced between each iteration until they meet

ALGORITHM 1 RIDGE GROWING

```
DetectRidges:
    sort grid into array by p.M in descending order
    CurrSeed = 0.99, CurrRidge = 0.98
while CurrSeed > SeedThresh || CurrRidge > RidgeThresh
    for each open end in RidgeList
        RidgeList_i = Trace(RidgeList_i)
    for each p_i where p_i.M >= CurrSeed && p_i ∉ RidgeList
        RidgeList += Trace(p_i)
    reduce CurrSeed and CurrRidge
        or CurrSeed = SeedThresh, CurrRidge = RidgeThresh
return RidgeList

Trace(p_0):
    Local = local neighborhood of p_0
    trim Local for alignment with existing ridge (if any)
    sort Local by M in descending order
    if Local[0] M >= CurrRidge
        Ridge += Local[0]
        if Local[0] ∉ RidgeList
            Ridge += Trace(Local[0])
return Ridge
```

the target thresholds. For each iteration, existing ridges are first grown, then new seed points are identified. A seed point is the beginning of a new ridge, and is identified if it has a maxima score greater than *CurrSeed* and is not already contained within one of the ridges in *RidgeList*. When growing a ridge from a tip of an existing ridge or a seed, the next series of points along the ridge are identified by the recursive *Trace* function in Algorithm 1. The *Trace* function examines all points in the local neighborhood centered on the active point p_0 (i.e., a 3 x 3 x 3 cube). To prevent ridges from doubling back on themselves, the *Trace* function trims all points which would change the current direction of ridge growth by more than π/2 radians. If the strongest remaining point has a maxima score *p.M* greater than the threshold *CurrRidge*, it is added to the current *Ridge* and becomes the next active point in the growing process. The ridge is recursively grown until no point where *p.M* > *CurrRidge* can be found, or the ridge intersects a boundary or another ridge. If two growing ridges intersect they are merged and become a single ridge. *SeedThresh* and *RidgeThresh* are respectively initialized as 0.8 and 0.5 due to how local extrema are represented by the maxima score [33].

D. Feature Tracking

Automated feature tracking assists the user in following vortex features over time and through evolution events. The tracking can be continuous [6, 23] or discrete [45, 46, 47, 48]. However, the discrete tracking is persistent for numerical robustness to noise and for preventing drift [23]. Our approach is discrete, using a combination of physics-based position prediction [45, 46] and attribute correspondence [47, 48]. Prior work on attribute-based feature tracking has focused on tracking the feature as a whole via attributes based on an encompassing ellipsoid or internal points. Our approach instead tracks individual parts of the feature based on segments along the ridges. By tracking segments, this approach can accommodate complex evolution events and use these segments to inform the surface-based visualization.

A ridge segment r_i is defined as a set of n (> 1) consecutive points (p_1, ..., p_n) along a ridge. Each segment has four attributes: orientation ($\vec{r}_{i,O}$), intensity ($r_{i,M}$), position ($r_{i,P}$), and velocity ($\vec{r}_{i,V}$). Based on the aggregate values of these points, the attributes of the segment represent a vector, scalar, 3D coordinate, and another vector, respectively:

$$\vec{r}_{i,O} = p_n(x_n, y_n, z_n) - p_1(x_1, y_1, z_1)\,, \tag{7}$$

$$r_{i,M} = \frac{\sum_{i=1}^{n} M_c(p_i)}{n}, \tag{8}$$

$$r_{i,P}(x,y,z) = \left\{\frac{\sum_{i=1}^{n} x_i}{n}, \frac{\sum_{i=1}^{n} y_i}{n}, \frac{\sum_{i=1}^{n} z_i}{n}\right\}, \tag{9}$$

$$\vec{r}_{i,V}(u,v,w) = \left\{\frac{\sum_{i=1}^{n} u_i}{n}, \frac{\sum_{i=1}^{n} v_i}{n}, \frac{\sum_{i=1}^{n} w_i}{n}\right\}, \tag{10}$$

where x_i, y_i, z_i represent respectively the cartesian coordinate of the i-th point p_i of the segment, and u_i, v_i, and w_i are the velocity components of the same point. $M_c(p_i)$ is the correlated maxima score at the point. The "similarity" between two ridge segments, r_1 and r_2, is measured by $S(r_1, r_2)$ as:

$$S(r_1, r_2) = \frac{\sum_{i=1}^{N} T(A_i(r_1, r_2))}{N}, \tag{11}$$

$$T\big(A_i(r_1, r_2)\big) = \begin{cases} \dfrac{A_i(r_1, r_2)}{T_i}, A_i(r_1, r_2) \le T_i \\ \infty \quad\quad , A_i(r_1, r_2) > T_i \end{cases}. \tag{12}$$

$S(r_1, r_2)$ is an average of N attributes of interest between the segments, $A_i(r_1, r_2)$ is the difference in the i-th attribute, while T_i limits and normalizes this difference as $T\big(A_i(r_1, r_2)\big)$. As $S(r_1, r_2)$ approaches 0, the similarity between the ridge segments r_1 and r_2 increases. This needs the i-th attribute $A_i(r_1, r_2)$ to be within its limit T_i. If this attribute exceeds T_i, the segments are considered dissimilar enough not to be tracked with $S(r_1, r_2) = \infty$. That is, T_i sets an upper boundary of dissimilarity between the segments for the i-th attribute correspondence. A larger value of T_i means a more flexible pairing of the segments. For feature tracking, r_1 and r_2 belong to time instants t_n and t_{n+1}. For this tracking we have used N=3 attributes: orientation, intensity, and expected position. The differences in orientation $O(r_1, r_2)$ and intensity $I(r_1, r_2)$ are defined as:

$$O(r_1, r_2) = \pi - \cos^{-1}\frac{\vec{r}_{1,O} \cdot \vec{r}_{2,O}}{\|\vec{r}_{1,O}\|\|\vec{r}_{2,O}\|}, \tag{13}$$

$$I(r_1, r_2) = \left|r_{1,M} - r_{2,M}\right|. \tag{14}$$

The difference in expected position is defined as:

$$P(r_1, r_2) = \left\|(r_{1,P} + \vec{r}_{1,V} * dt) - r_{2,P}\right\|, \tag{15}$$

where dt is the time between t_n and t_{n+1}. This Euler projection is sufficient for low values of dt. The projection is based on two motion components of a vortex feature: the rotation of the vortex itself and the motion of the feature as a whole [45, 46]. Because the center of the vortex rotates minimally, its velocity is dominated by the motion of the feature as a whole. This velocity is thus available for predicting the future position of a ridge segment. Points of the segment are improbably at the exact location of zero rotation. In turn, the average velocity of these points $\vec{r}_{i,V}$ approximates the motion of the segment. This

ALGORITHM 2 FEATURE TRACKING

```
TrackRidges:
  // Segment ridges
  for each set of ridges R_i in each time instant t_j
    segment R_i into r_{i,1} .. r_{i,n}
  // Primary segment tracking
  for each segment r_i in each time instant t_j
    candidate = Track(r_i , t_{j+1})
    if candidate.past = 0 or S(r_i, candidate) < S(candidate, candidate.past)
      r_i.future = candidate, candidate.past = r_i
  // Fill gaps based on the nearest neighbor
  for each ridge in R_i
    for each ridge segment r_{i,j} in R_i
      if r_{i,j}.past ≠ 0 and r_{i,j±1}.past = 0
        r_{i,j±1}.past = r_{i,j}.past
      if r_{i,j}.future ≠ 0 and r_{i,j±1}.future = 0
        r_{i,j±1}.future = r_{i,j}.future

Track(r_i, SegList):
  // Limit scope based on proximity
  for each r_j in SegList where T(Attr_P(r_i, r_j)) < 1.0
    ShortSegList += r_j
  // Identify strongest match
  for each r_j in ShortSegList
    r_j.S = S(r_i, r_j)
  sort ShortSegList by S in ascending order
  if (ShortSegList[0] < 1.0)
    return ShortSegList[0]
```

projection computes one predictor, compared to multiple predictors/correctors [46].

The limit T_i of the i-th attribute correspondence is necessary to capture terminal evolution events (i.e., creation and dissipation) in discrete feature tracking [47] and to prevent extraneous calculations. The orientation attribute limit is initialized as $T_O = \pi/4$, half of the largest angular difference between two segments. The intensity attribute limit is set as $T_I = 0.1$, permitting a 10% change in intensity between time instants. The proximity attribute limit is $T_P = d/2$, where d is the length of a ridge segment. This is the smallest distance which ensures that even if the segment distribution along the length of the ridge is maximally offset, $r_{2,P}$ will still be within this limit. When examining the attribute correspondence between r_1 and a pool of potential candidates, the ridge segment r_2 that provides the smallest similarity measure $S(r_1, r_2)$ and $S(r_1, r_2) \neq \infty$ is the best candidate for the tracked future version of r_1.

Algorithm 2 describes the procedure of feature tracking via the function *TrackRidges* with 3 phases: ridge segmentation, preliminary tracking, and gap filling. Ridge segmentation divides all ridges R_i at t_i into a set of $r_{i,1} \ldots r_{i,n}$ segments. All segments are roughly the same size and each ridge has on average 10 segments. Preliminary tracking is performed through the function *Track* between a segment in time t_i and each of the segments in t_{i+1}. If multiple segments identify the same future segment, the one with the smallest similarity measure is used. By letting the individual segments track independently, this approach can capture complex evolution events with no additional steps or bi-directional tracking [5]. A bifurcation is found when segments along a single ridge identify two or more different ridges as their future versions, while an amalgamation is the inverse. Creation and dissipation events are found when segments along a ridge do not have a past or future version respectively. Feature interactions are identified by topological changes of their ridges due to deflections or evolution events.

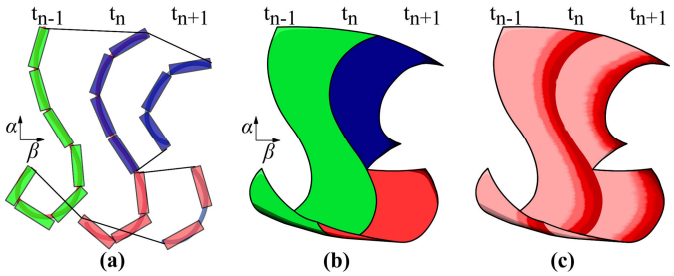

Fig. 3. Evolution surface generation after tracking: (a) segment grouping based on similarities in tracking, (b) surface generation to produce three sub-surfaces based on the bifurcation between times t_n and t_{n+1}, and (c) the final surface with motion waves.

Primary tracking identifies a past and future version for most of the segments. Segments which did not identify a past or future version copy the results of their nearest neighbor along their ridge to fill in gaps. The remaining unmatched segments represent creation or dissipation events. This additional step ensures large regions of contiguous surfaces, and smooth surface separations at evolution points.

E. Surface Visualization

Feature evolution is visualized via a set of quadratic non-uniform rational B-spline (NURBS) surfaces. NURBS surfaces approximate between a set of discrete control points and can be used in curve fitting [49]. This approximation provides two benefits: to interpolate the discretized 4D grid of the flow data closer to the continuous spatiotemporal positions of the flow features, and to intuitively present the continuity of feature motion over time. As a degree-2 (i.e., quadratic) approximation, such NURBS surfaces were adequate in our earlier work [27].

In brief, a NURBS surface is comprised of a rectilinear grid of control points in the α and β directions, where $0 \leq \alpha, \beta \leq 1$ as commonly used. The control points in the α direction are the set of sequential points along the tracked ridge at a given time. The control points traversing along the β direction represent this ridge as it exists in other times, as indicated in Fig. 3(a). The position of a given point $p(\alpha, \beta)$ on the NURBS surface is found via the contributions of control points on the ridges from de Boor's algorithm with an open uniform knot vector [50].

Since a NURBS surface traditionally uses a rectilinear set of control points, multiple sub-surfaces are used to represent bifurcation and amalgamation. Figure 3 depicts the procedure of generating these sub-surfaces, including a bifurcation event. To define the control points of a sub-surface along the α direction, consecutive ridge segments are grouped together if they share the same past and future ridges. As depicted in Fig. 3(a), the ridge segments at t_n belong to two such groups (red and blue) according to their association with the two ridges at t_{n+1}. Sub-surfaces are formed in the β direction according to tracking between segment groups in different time instants. Cases where a single ridge is comprised of two or more groups indicate an evolution event. Such an event ends sub-surfaces in the β

TABLE 1 FLOW DATA PROPERTIES

Dataset	Dimensions (x × y × z × t)	Reynolds Number (*Re*)	Uniform Grid
Pyramid	31×33×26×51	28000	YES
Plate	147×87×32×100	1200	NO

direction and new sub-surfaces begin, as presented in Fig. 3(b). Currently, each sub-surface is approximated independently for a C^0 continuity to reduce computation time.

Vortex features are differentiated via coloration; all sub-surfaces representing a feature share the same color. To provide the user with context of a feature's velocity and shape, a wave-like pattern is applied to the surface texture [20, 51, 52]. This pattern is shown in Fig. 3(c), and is tied to both a user-defined period t_p and RGB color vector $c\{r, g, b\}$. At time t, the color of any surface point $p_C(t)$ is calculated by:

$$p_C(t) = c + \frac{(\{1,1,1\} - c)(t \bmod t_p)}{t_p}. \quad (16)$$

This yields a linear wave-like pattern on the evolution surface between white $\{1, 1, 1\}$ and the user-specified color c, though other patterns could be explored [52]. The leading edge of these waves implies the direction of travel, and the distance between waves reveals the speed of travel. These surfaces are rendered semi-transparently to allow visualization of their overlaps or intersections for analysis. This rendering and transparency were performed using the open-source Visualization Toolkit (VTK) [53]. The built-in VTK algorithm of depth peeling was used for the transparency, with a maximum depth of 20 peels to preserve smooth user interaction.

For larger datasets it is not possible to render evolution surfaces for all features simultaneously without high visual clutter. Therefore, after ridge extraction the user selects specific features of interest to render as evolution surfaces. The ridges along each surface that are active at user-specified time instants are also rendered. These ridges assist in visualizing the temporal relationship between features, and provide context to other visualization techniques (e.g., isosurfaces) that may be used in conjunction with the evolution surfaces. To aid in depth perception, the user could adjust his/her view and has access to an optional stereoscopic display.

IV. RESULTS

We applied the evolution surface approach to visualize two unsteady wakes provided by our collaborators in fluid dynamics. Table 1 indicate the properties of the datasets: the pyramid and the plate. These datasets were examined due to prior work of our collaborators, allowing their verification of the evolution surface approach. Moreover, the features of interest in these datasets underwent smooth translational motions, making them suitable for the maxima score approach (i.e., Galilean invariant vortex detection). The pyramid dataset was collected through empirical measurement, whereas the plate dataset was generated through computational fluid dynamics (CFD) simulations. Both empirical measurement and CFD represent the primary methods of acquiring flow data in practice. Together, these datasets represent disparate conditions regarding turbulence intensity, grid uniformity, scale, and uncertainty. The disparity between these datasets has allowed us to test the versatility of our approach and make meaningful comparisons to other techniques.

A. Pyramid Flow

1) Data Description: This dataset is the wake of a pyramid-shaped obstruction (base d^2 = 45 mm × 45 mm, height h = 39 mm) positioned upstream of the measured volume, as illustrated in Fig. 4(a). The dataset was empirically acquired in prior work [54]. The acquisition utilized Stereoscopic Particle Image Velocimetry, a non-intrusive technique for obtaining velocity data of a fluid flow. Of interest to our collaborators were the interactions between vortices, and the behavioral differences among disparate vortex shedding "cycles".

The pyramid flow contained four periodic vortex features of interest, two of which are presented in Fig. 4(b). There are two kinds of vortices present in this wake: the "tip vortex" which sheds from the tip of the pyramid and the "base vortex" which sheds from where the pyramid base meets the mounted surface. These features are mirrored about the midline labelled in Fig. 4(a); the right tip is paired with the left base and vice versa. The maxima score ridge lines of these features are highlighted and labelled in Fig. 4(b). Each individual feature existed in the measurement volume for about 9 time instants. The flow had distinct shedding cycles, corresponding to its turbulence energy levels. We examined the "high cycle" containing intense vortex

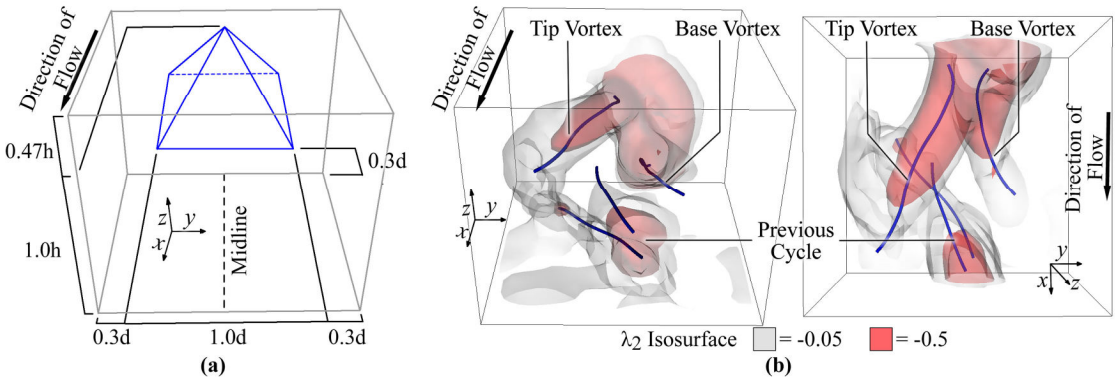

Fig. 4. The properties of the high cycle pyramid dataset (a) dimensions of the flow field and pyramid location [d^2 and h are the base surface and height of the pyramid, respectively], and (b) visualization of the vortices using λ_2 isosurfaces and maxima score ridges.

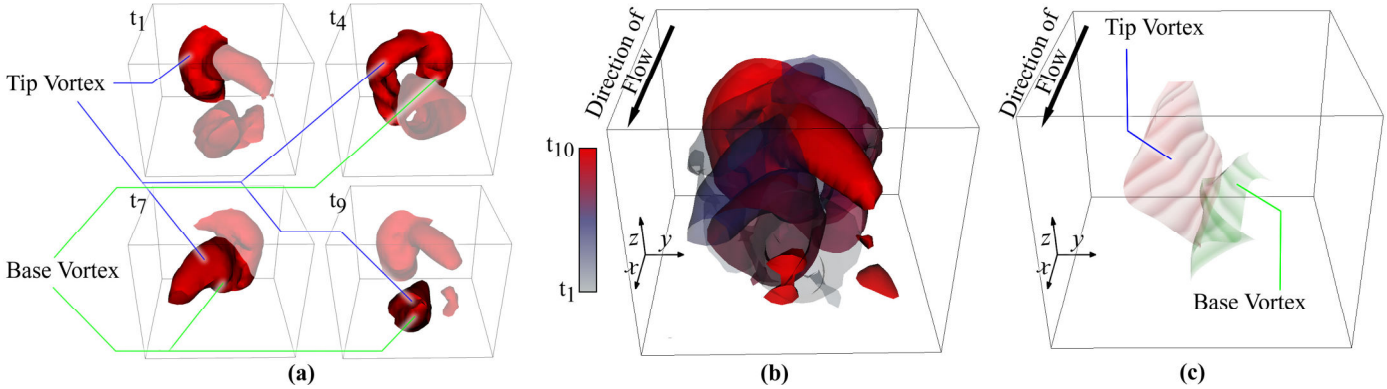

Fig. 5. Visualizing the spatiotemporal movement of a tip and base vortex in the pyramid wake using (a) still images, (b) a chronovolume, and (c) evolution surfaces.

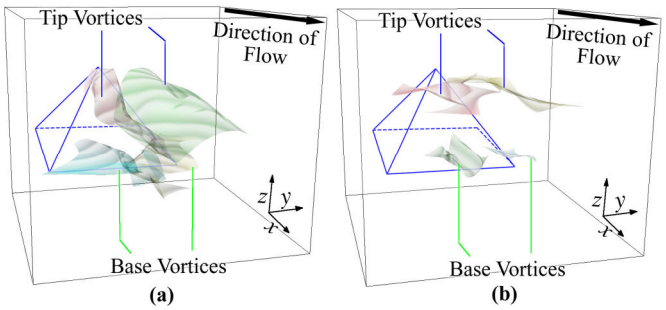

Fig. 6. Evolution surfaces of the four vortex features in the pyramid flow during two different shedding cycles: (a) the high cycle; and (b) the low cycle

activity and periodic shedding, and the "low cycle" which had abated vortex activity and unpredictable shedding.

2) Visualization: Fig. 5 depicts a single tip/base pair of vortices shedding in the high cycle visualized using still images, a chronovolume, and two evolution surfaces. A qualitative look at these visualizations highlights some of the challenges posed by this dataset to still images and chronovolumes. For Fig. 5(a), space constraints limit the temporal resolution available to the still images. Combined with the irregular deformation of the isosurfaces, the continuity of the feature behavior between time instants is unclear (e.g., what changes occur between t_4 and t_7). The chronovolumes in Fig, 5(b) covers the same time interval as the still images and renders at half the temporal resolution of the dataset with similar 4 time instants overlaid. Despite the temporal resolution reduction, the relative size of the vortex features causes occlusion and prevents a clear view of individual time instants. Additionally, the volumetric isosurfaces suffer due to the intensity difference and proximity of the base and tip vortices, making them difficult to visually separate. In contrast, the ridge extraction of evolution surfaces permits the base vortex, the weaker of the two, to be clearly visible. Figure 5(c) illustrates the shedding behavior of these features with minimal occlusion for the same interval. What occurs between t_4 and t_7 is the extension of the tip vortex away from the pyramid, and the part closest to the pyramid dipping towards the mounted surface. The vortex then separates from the obstruction as it crosses the midline and exits the volume at the face nearest to the viewer. The base vortex can be seen extending part-way up the volume, shedding and re-orienting itself towards the y-axis where it interacts with the tip vortex. These interactions fit the behaviors expected by our fluid dynamist collaborators, though they could not be observed using the still images or chronovolume in Figs. 5(a) or (b).

The compact visualization provided by evolution surfaces also makes it capable of illustrating broader changes in feature behavior spanning multiple shedding cycles. Figure 6 presents the evolution surfaces of all four vortex features of interest during both high and low shedding cycles. As depicted in Fig. 6(a), there is a prominent dip in both tip vortices as they shed, which leads to the previously noted interaction with the base vortices. Figure 6(b) shows that the low cycle tip vortices lack this dip, and thus the base vortex interaction which results in the midline crossing and orientation change does not occur. When inspecting the isosurfaces of these shedding cycles using still images, a difference in behavior was observed but the nature of the change in the tip-base vortex interaction was not visible.

3) Verification: The accuracy of these surfaces was confirmed with our collaborators (fluid dynamicists) in two stages. Firstly, the main part of ridge positions and shapes in several time instants were confirmed by using λ_2 isosurfaces, as shown in Fig. 4(b), and by the reported outcomes on the same dataset in prior work [54]. Secondly, evolution surfaces of these ridges were examined by comparisons to still images and animations of this flow. This ensured that the surfaces correctly matched the movements of the features. No significant deviations were noted for any feature over the time series (approximately 6 shedding cycles for each feature).

B. Plate Flow

1) Data Description: The obstruction of the plate flow was a thin, flat plate with zero width (x), a finite height (y), and an infinite length (z). Figure 7(a) presents a representative section of the flow field along the length of the plate. The dataset was generated through CFD simulation on an adaptive rectilinear grid, biased with greater point density around the plate [55]. Of interest to our collaborators was the behavior of vortex features in the region near the plate, and the way this influenced vortex shedding downstream.

The challenges presented by the plate dataset to visualization are different to those in the pyramid. Relative feature movement between time instants is smaller than in the pyramid dataset, though the features are numerous and undergo more complex evolution events. The features behind the plate are densely packed and have several magnitudes greater intensity than those

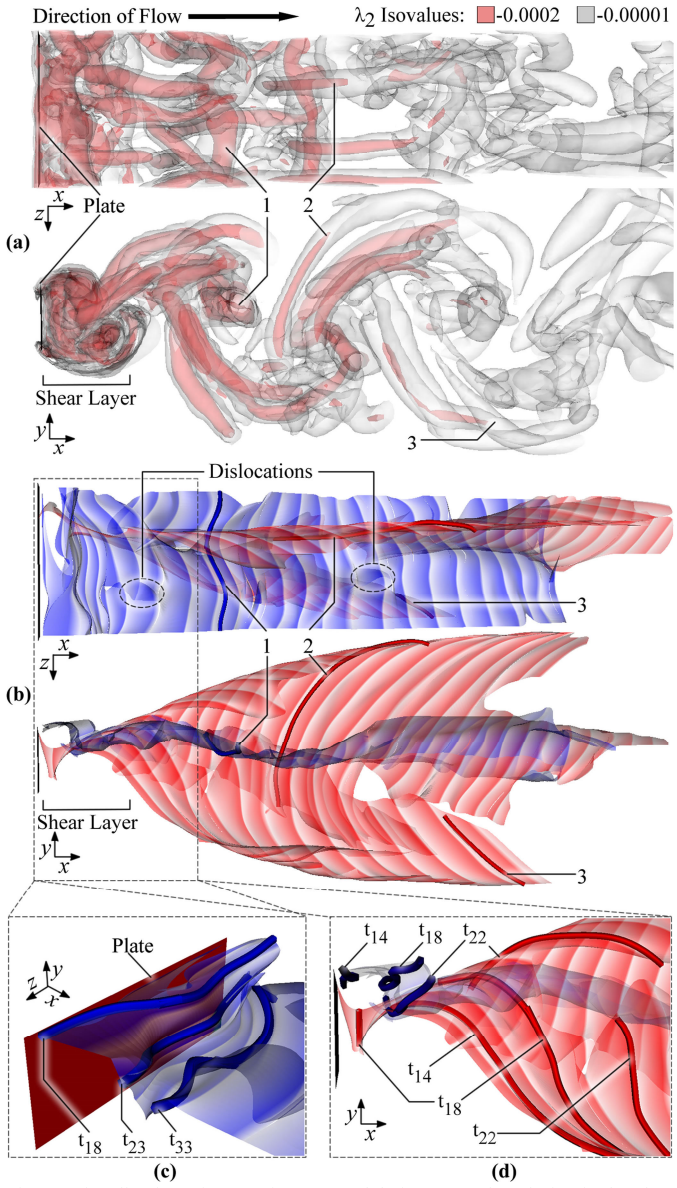

Fig. 7. Visualization of vortex features and their temporal evolution in the plate flow: (a) feature visualization using λ_2 isosurfaces; (b) evolution surface visualization of the labelled features 1 (blue), 2 (red), and 3 (red) with their highlighted ridges at the same time instant; (c) a close-up of feature 1 forming near the plate; and (d) a close-up of these features interacting near the plate. [Highlighted ridges in (c) and (d) range from time instant t_{14} to t_{33}.]

downstream. There are two main types of vortex features present in this flow: "rollers" (labelled 1 in Fig. 7) that rotate along the z-axis and form from the shedding of the shear layer and "ribs" (labelled 2 and 3 in Fig. 7) that align to the x-y plane and wrap around the rollers. The rotation of the rollers causes bifurcation and amalgamation activities. These challenges combine to make analysis through still images time consuming, and the relationships between features over time uncertain.

2) Visualization: The evolution surface approach provides three main benefits to analyzing this flow field. First, a ridge-based feature definition permits identification of individual vortices within the region near the plate, as well as consistent feature extraction across the entirety of the volume. Second, the evolution surface provides a view of a feature's behavior over long timespans. This allows for examination of how the feature formed, and what impact that has on the feature later downstream. Third, these surfaces can concisely portray the relationships between features that may have bifurcated or amalgamated at distant points in time.

Figure 7(b) gives the evolution surface representation of the user-selected features – labelled as 1, 2, and 3 and highlighted at the same time instant. The roller (1, blue) and ribs (2, 3; red) shed from the shear layer and travel downstream. These features are spatially separate at the highlighted time instant. A close-up of the formation of the roller 1 is depicted in Fig. 7(c) with three time instants highlighted. The roller seems to form from the recirculation of the shear layer towards the plate before travelling downstream. This roller merges with other ridges into a single surface at the highlighted time instant, as indicated in Fig. 7(b). This amalgamation occurs as the roller exits the shear layer, while influence from nearby features is reduced.

The evolution surface approach also allows the identification of some seldom and complex vortex dynamics, manifested as the darker patches circled in Fig. 7(b). Each patch occurs at the location of a vortex dislocation event, which signifies the brief split of a single roller into two separate rollers. The core lines of these two rollers coexist in close proximity for a very short time. These core lines rotate together about a common center by the Biot-Savart induction principle [8, 28, 31]. Shown by an overlap between sub-surfaces (i.e., a dark patch), such a dislocation event is generally preceded by a strong distortion of the roller. This distortion is consistent with the observed deformation of the roller in t_{33}, as exemplified in Fig. 7(c).

Travelling downstream, the ribs 2 and 3 change their orientations and bifurcate at the boundary of the sampled volume, as depicted in Fig. 7(b). What is unapparent without feature tracking is an earlier bifurcation to form these ribs. As seen in Fig. 7(d), this bifurcation occurs near the plate between t_{18} and t_{22} from one single ridge at t_{14} and results from interaction between the ridge and a roller (not shown because it obscures the bifurcation). Just before t_{18}, there is a third bifurcation from the single ridge interacting with the formation of the roller 1. This bifurcation produces two ridges at t_{18}. The larger one extends downstream to form the ribs 2 and 3, as described. The smaller one recirculated back to strike the plate. Importantly, this third bifurcation occurs within the shear layers and is not easily visible through isosurfaces.

3) Verification: The numerous features in the plate dataset proved useful for testing the accuracy of our approach, providing a variety in feature strengths, proximities, and evolution events. To evaluate the ridge extraction and feature tracking algorithms, we correlated the results of these algorithms to those obtained manually from an expert user familiar with the dynamics of the flow. The user's manual results were based on a combination of still images and animation. Forty vortex features at time instant 50 were identified using the λ_2 criterion isosurfaces, comprised of a mix of rollers (5), ribs (27), and features forming near the plate (8). Our evaluation focused on the following criteria: *completeness* of the algorithm results (does the ridge/surface capture the

TABLE 2 EVALUATION OF EVOLUTION SURFACE ALGORITHMS

Algorithm	Complete	Correct
Ridge Extraction (t_{50})	40 (100%)	40 (100%)
Ridge Extraction ($t_{1\text{-}100}$)	38 (95%)	38 (95%)
Feature Tracking	40 (100%)	39 (97.5%)

entirety of the feature) and *correctness* in matching with the manual results (does the ridge/surface differentiate between nearby features). Since the accuracy of the ridge extraction impacts the evolution surfaces, we distinguished between individual ridges (t_{50}) and all ridges within a surface ($t_{1\text{-}100}$). Note that this second evaluation is not on individual ridges, all ridges within the surface passed or failed together.

Table 2 presents the results of this comparison. The major deficiency noted was in the ridge extraction in the regions around the rollers over the lifetime of the features. We noted two features which were missing a connective ridge, one of which led to an unexpected termination and the other which tracked a nearby parallel feature (recorded as the correctness failure for feature tracking). Both cases happened at the boundary during interaction with the rollers. Other cases of gaps in the ridges at individual time instants did not negatively impact the resulting surface. We also noted two instances where low intensity of the ribs resulted in unexpected merges between ridges adding geometry to the evolution surfaces, which would impede analysis. Despite this, the ridge extraction was successful at identifying features within the region near the plate, a task that was time-consuming to perform manually due to occlusion and intensity variation. Our automated feature tracking greatly outperformed manual tracking when it came to evolution events. Small vortices resulting from bifurcations near the rollers were often missed by the user due to occlusion.

C. Comparison

1) Visual Complexity: The small number of features presented in the pyramid dataset made it possible to isolate the isosurfaces of features within a shedding cycle. This permitted us to examine the visual complexity of the same features by the evolution surface, compared to the still images and chronovolumes. This comparison was conducted using three common measures of visual clutter [56]. Aiming to describe how humans interpret a visual display, these measures quantify the prominence of features as feature congestion, the variety of the display as subband entropy, and the numerosity of features as edge density. Higher values in these measures indicate higher clutter, and have shown a direct relationship to increased time to perform visual analysis tasks [56, 57]. For visualization, clutter reduction is not a direct measure of quality; but desirable as it improves user search times during visual analyses.

We applied these measures to 15 different viewing angles for each of the three visualization techniques; aligned with each face and corner of the bounding box, as well as the oblique angle in Fig 5. For the still images, all 4 time instants were tiled together into one image as shown in Fig. 5(a). This tiled image was resized to match the dimensions of both the chronovolume and evolution surface images. The resize is necessary to prevent size from influencing the comparison. We also measured the empty cube outlining each visualization to provide a baseline of the metric's overheads. The clutter metrics for the viewing angles conformed to a normal distribution, so the one-way ANOVA and Tukey Test were used to verify that the visualization techniques were statistically distinct. The ANOVA results are given in Table 3, each pair produced a Tukey Test HSD $p < 0.01$ confirming that there is a statistical difference between the visualization methods. The means of the visualization techniques and the empty cube are provided in Table 4. Under all three metrics, evolution surfaces yielded the least cluttered display. This enabled the display of all four features in a full shedding cycle simultaneously, as shown in Fig. 6(a). A display of the same features would require more space using still images or further occlude the chronovolume.

TABLE 3 ONE-WAY ANOVA FOR CLUTTER METRIC RESULTS

Metric	ANOVA
Feature Congestion	$F(2, 28) = 337.79$ $p < 0.0001$
Subband Entropy	$F(2, 28) = 249.73$ $p < 0.0001$
Edge Density	$F(2, 28) = 54.79$ $p < 0.0001$

TABLE 4 CLUTTER METRICS OF VISUALIZATION TECHNIQUES

Visualization	Feature Congestion	Subband Entropy	Edge Density
Still Images	4.934 ± 0.208	3.018 ± 0.053	5.556 ± 0.237%
Chronovolume	3.392 ± 0.135	2.400 ± 0.046	4.351 ± 0.163%
Evolution Surfaces	3.030 ± 0.129	2.275 ± 0.070	3.600 ± 0.261%
Empty Cube	2.706 ± 0.094	1.795 ± 0.010	1.950 ± 0.096%

mean±standard error

The long lifetimes of the numerous features in the plate dataset required still images in an adverse quantity and produced a heavily occluded chronovolume. Both were unsuitable for user analysis, compared to that from the evolution surface as shown in Fig. 7(b). Thus, the plate dataset was excluded from the visual complexity comparison.

2) Ridge Extraction: In Fig. 8 we have compared our ridge extraction results to three other methods. On the plate dataset, Fig. 8(a) compares the volumetric results of the correlated maxima score and the Frangi filter [39] (a Hessian-based ridge detector used in medical imaging to highlight fine tube-like vasculatures). While both detectors found many of the same features, the Frangi filter struggled with the more diffuse features circled in Fig. 8(a). Figure 8(b) compares our ridge extraction (blue) to the vortex core extraction implemented in the commercial visualization toolkit Amira (light red). Amira's vortex cores are based on the PV of velocity and vorticity [35]. These PV lines captured only a small subset of features compared to our ridge extraction. Figure 8(c) presents a comparison between our ridges and the core lines based on extrema extracted using the Q criterion (i.e., Q extrema) [40], another Galilean invariant vortex detector [32]. There is an overlap between the ridges extracted by the maxima score and those identified by Q extrema, although some minor differences exist between the end-points of the ridges and of the core lines.

In addition, Figs. 8(d) and 8(e) reveal results of the Frangi filter and Q extrema respectively on the pyramid dataset, compared to those of the maxima score. The Frangi filter and Q extrema fail to identify the features in the circled regions. The

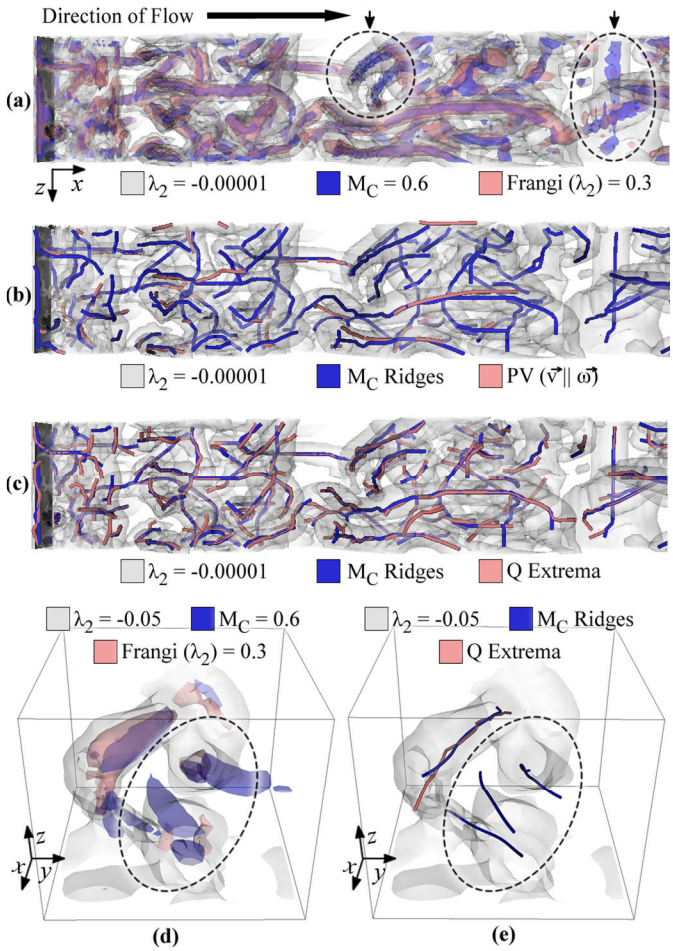

Fig. 8. Vortices in the plate dataset visualized by the λ_2 criterion comparing (a) correlated maxima score to the Frangi filter, (b) maxima score ridges to velocity || vorticity PV lines, and (c) *Q* extrema lines. The vortices of the pyramid were also examined using (d) the Frangi filter and (e) Q extrema lines

failure might come from two factors: the increased uncertainty in this dataset due to its empirical measurements and the proximity of the features to the boundary of the flow field. These factors inhibit the identification of tube-like configurations and critical points necessary for the Frangi filter and Q extrema, respectively. In contrast, the maxima score accommodates these factors to yield all ridges of vortices. The outcome of the PV is only a part of one feature and, in turn, is not included in Fig. 8.

3) Performance: All computations were performed on a workstation with an Intel Xeon 35-1620 CPU at 3.6 GHz. Table 5 presents the computation time for each step of the approach for both datasets. The processing was performed on a single thread, though this approach is suitable for multithreading as each time instant is handled independently for all processing steps. Likewise, the vector data for each time instant only needs to be loaded once, after ridge extraction only the points belonging to ridges are required. As expected based on prior work [33], the computation time was primarily accrued from the maxima score correlation (which included initial feature extraction). After tracking, feature selection and surface visualization occur in real-time for user interactivity.

In comparison to advection field based spatiotemporal

TABLE 5 COMPUTATION TIME (SECONDS)

Dataset	Maxima Score Correlation ($\lambda_2 + \omega$)	Ridge Extraction	Feature Tracking
Pyramid (8.3) [a]	3.063 (1.803)	0.129	0.007
Plate (167.2) [a]	229.233 (117.562)	8.736	4.302

[a] Average number of ridges per time instant

TABLE 6 COMPUTATION TIMES OF TEMPORAL TRACKING

Tracking Method	Dimensions (x,y,z,t)	# of Points	CPU Speed	Time (s)	3D Features
Evolution Surfaces	147x87x32x100	40.9m	3.60Gz	242	Yes
	31x33x26x51	1.4m	3.60Gz	3	Yes
3D PV FFF [6]	256x337x65x---	---	---	---	Yes
Combinatorial FFF [58]	256x256x1x256	16.8m	2.40Gz [a]	149	No
Stable FFF [23]	256x256x1x256	16.8m	2.40Gz [a]	333	No
Advected Tangent	228x100x1x250	5.7m	2.66Gz	52	No
Curves [7]	192x64x48x102	60.2m	2.66Gz	9600	Yes

[a] Two processors, 4 cores each

surfaces, such as with FFFs, our approach is computationally light due to the lack of higher order derivatives. As no computation times have been reported for 3D feature tracking fields [6], Table 6 provides times for 2D datasets in prior work. To accommodate the difference of workstations indicated above and reported in the prior work, the CPU clock speed of each workstation is included where available. When accounting for the CPU speed and data dimensions, our 3D evolution surface approach is roughly comparable to times reported for tracking 2D features using FFFs [58]. However, the cost of increasing dimensions for these calculations is non-linear due to larger matrices and higher order calculations needed. While not explicitly a feature-tracking field, advected tangent curves produce 3D or 4D fields for spatiotemporal streak lines/surfaces [7]. As shown in Table 6, the advected tangent curves yields an order of magnitude increase (~20 times after taking account dimensional differences) in computation time when extending the z-axis from 1 to multiple. Meanwhile, our feature tracking is the least time intensive operation in our algorithm, and scales in $O(n^2)$ time to the number of ridges extracted n.

V. DISCUSSION

The evolution surface approach provides a concise visualization of spatiotemporal evolution of flow features. Though many of the stages in this approach build upon prior work, in combination they enable a final visualization with novel capabilities. Evolution surfaces do not increase in visual complexity with higher temporal resolutions, allowing for more accuracy in locating evolution events in space and time than with still images or chronovolumes. Their low visual complexity is useful for rendering long time series, or multiple features simultaneously. This allows identification of the relationships between temporally distant features or events, as well as overall behavioral patterns in the flow. Evolution surfaces can be used to clearly compare disparate flow cycles (as in Fig. 6), or to characterize flow behavior under different conditions. In addition, the flexibility of the segment-based tracking enables tracking and visualization over potential feature evolution events [5]. Our collaborators in fluid

dynamics have expressed interest in the approache's capabilties at visualizing feature interactions within regions of high turbulence (e.g., near the plate). The ability of condensing feature behaviors during long periods into a single display would be well suited for exploratory or comparative analyses. Besides, the evolution surface approach is less time consuming than the methods listed in Table 6.

To accompany the tracking and rendering of evolution surfaces, we've also presented a computationally light ridge extraction scheme. Our maxima score-based ridge extraction was compared favorably to the Frangi filter, PV lines, and Q extrema ridge detectors on the plate and pyramid datasets. The cause of the discrepancy in ridges detected by PV lines compared to maxima score ridges in Fig. 8(b) may be due to several factors including grid density, Reynolds number, and reference frame invariance. Though similar results were produced on the plate, Q extrema suffered in the pyramid dataset due to the critical points (i.e. the seed points) having exited the flow field at the sampled time instant. Regardless, an interesting property of our tracking and surface rendering is that it can be adapted to different ridge extraction techniques with minor modification. For instance, we have applied this tracking to the PV and Q extrema in Fig.8 by adjusting the tracking intensity attribute. Rather than using the maxima score, intensity was replaced by the λ_2 criterion (or Q) to produce evolution surfaces based on the PV vortex cores. The independence between ridge extraction and the final surfaces means that as new definitions of vortices are explored, evolution surfaces will remain viable for spatiotemporal rendering with minimal computation costs. Although this paper has focused on vortices, evolution surfaces may be applied to other tube-like flow structures in the future.

In comparing our tracking and rendering results to prior work, advection through FFF and PV produced the most similar surface-based visualization [6]. However, a known limitation of these PV-based surfaces is an inability to model fold bifurcations, as PV fold bifurcations do not exist and PV pitchfork bifurcations might be unstable [6]. This is a non-trivial problem for other stream surface techniques as well, due to the similarity in advected particle behavior between bifurcations and size change. Our tracking technique resolves this by tracking from ridge to ridge, producing a reference point at each time instant. In combination with the segment-based tracking, this accommodates the more complex evolution events that occur in more turbulent flow data. In prior work we have even used a modified version of this approach to examine the wake of a train with a Reynolds number Re = 86,000 [27].

Despite these benefits, there are some tasks in which evolution surfaces are not suitable. While the ridge abstraction reduces visual complexity, it hides the relative size of features. It may be possible to encode this information into the surface through color in the future [52], for now other techniques (e.g., isosurfaces) can be used alongside evolution surfaces when needed. There is also a strong reliance on the accuracy of the ridge extraction technique. Being region-based, the maxima score approach may find extrema close enough to – but not at – the "true" locations of vortices. The effect of this "regularization" on topology of vortex features would be an interesting investigation in the future. The maxima score ridge extraction was noted to have inconsistent performance in locally diffuse features (such as around the rollers in the plate dataset) which negatively impacted the final surfaces. The Hessian-based ridge detection and Q extrema core lines also struggled in this area; finding a ridge extraction scheme which can resolve this problem is left to future work. This ridge extraction also identified seldom-occurring vortex dislocation events. Confirming such events is relegated to future work however, because the confirmation requires a detailed analysis of the corresponding vorticity field. The existence of such events also motivates further robustness testing of the feature tracking and examining NURBS approximation errors in future investigations. Evolution surfaces can enable analysis of user-selected features for turbulent flow fields with numerous vortices (e.g., the plate dataset). However, this visualization would suffer from occlusion if the surfaces of all vortices are rendered simultaneously. Other future work is to explore how to present these surfaces for occlusion and clutter reduction.

VI. Conclusions

We presented a novel approach for visualizing the temporal evolution of flow features. Evolution surfaces allow user analysis of multiple features simultaneously while better controlling the visual complexity of the image compared to existing techniques. This offers a new way of visualizing spatiotemporal interactions of features, yielding a potential means of comparing multiple features across datasets.

Acknowledgement

The authors would like to thank Dr. D. Wood at the Dept. of Mechanical and Manufacturing Engineering, University of Calgary; and Dr. A. Hemmati at the Dept. of Mechanical Engineering, University of Alberta for providing us with flow data, feedback, and expertise in the field of fluid dynamics. We would also like to acknowledge CMC Microsystems for the provision of the Amira visualization toolkit.

References

[1] W. Aigner, S. Miksch, W. Müller, H. Schumann, C. Tominski, "Visual methods for analyzing time-oriented data," *IEEE Trans. Vis. Comput. Graphics*, vol. 14, no. 1, pp. 47-60, 2008.

[2] B. Bach, P. Dragicevic, D. Archambault, C. Hurter, S. Carpendale, "A descriptive framework for temporal data visualizations based on generalized space-time cubes." *Comput. Graph. Forum*, vol. 36, no. 6, pp. 36-61, 2017.

[3] B. Tversky, J.B. Morrison, M. Betrancourt, "Animation: can it facilitate?" *Int. J. Hum.-Comput. St*., vol. 57, no. 4, pp. 247-262, Oct 2002.

[4] D.H. Laidlaw *et al.*, "Comparing 2D vector field visualization methods: a user study." *IEEE Trans. Vis. Comput. Graphics*, vol. 11, no. 1, pp. 59-70, 2005.

[5] R. Samtaney, D. Silver, N. Zabusky, J. Cao, "Visualizing features and tracking their evolution," *Computer*, vol. 27, no. 7, pp. 20-27, 1994.

[6] H. Theisel, J. Sahner, T. Weinkauf, H.-C. Hege, H.-P. Seidel, "Extraction of parallel vector surfaces in 3D time-dependent fields and application to vortex core line tracking," *VIS '05,* Minneapolis, MN., pp. 631-638, 2005.

[7] T. Weinkauf, H.-C. Hege, H. Theisel, "Advected tangent curves: a general scheme for characteristic curves of flow fields," Comput. Graph. Forum, vol. 31, no. 2.4, pp. 825-834, 2012.

[8] G. Haller, "An objective definition of a vortex," J. Fluid Mech., vol. 525, pp. 1-26, 2005.

[9] G. Haller, A. Hadjighasem, M. Farazmand, F. Huhn, "Defining coherent vortices objectively from the vorticity," J. Fluid Mech., vol. 795, pp. 136-173, 2016.

[10] G. Tadmor et al., "Reduced order models for closed-loop wake control," Philos. Trans. R. Soc. A, vol. 369, no. 1940, pp. 1513-1524, 2011.

[11] A. Orellano, S. Sperling, "Aerodynamic improvements and associated energy demand reduction of trains," in The Aerodynamcis of Heavy Vehicles II: Trucks, Buses, and Trains. Springer Berlin Heidelberg, pp. 219-231, 2009.

[12] N. Rott, "Note on the history of the Reynolds number," Annu. Rev. Fluid Mech., vol. 22, pp. 1-12, 1990.

[13] F.H. Post, B. Vrolijk, H. Hauser, R.S. Laramee, H. Doleisch, "The state of the art in flow visualization: feature extraction and tracking," Comput. Graph. Forum, vol. 22, no. 4, pp. 775-792, 2003.

[14] A. Brambilla, R. Carnecky, R. Peikert, I. Viola, H. Hauser, "Illustrative flow visualization: state of the art, trends and challenges," Eurographics 2012 – STARs, Cagliari, Italy, pp. 75-94, 2012.

[15] V.I. Arnold, "Sur la topolgie des écoulements stationnaires, des fluides parfaits," in C. R. Acad. Sci., vol. 261, 17-21, 1965.

[16] R. Ploetzner, R. Lowe, "A systematic characterisation of expository animations," Comput. Hum. Behav., vol. 28, no. 3, pp. 781-794, 2012.

[17] J. Woodring, H.-W. Shen, "Chronovolumes: a direct rendering technique for visualizing time-varying data," Eurographics: Volume Graphics 2003, Goslar, Germany, pp. 27-34, 2003.

[18] W.-H. Hsu, J. Mei, C.D. Correa, K.-L. Ma, "Depicting time evolving flow with illustrative visualization techniques," LNICST, vol. 30, pp. 136-147, 2010.

[19] T.-Y. Lee, H.-W. Shen, "Visualizing time-varying features with TAC-based distance fields," IEEE Pac. Vis. Symp. '09, Beijing, China, pp. 1-8, 2009.

[20] A. Joshi, P. Rheingans, "Illustration-inspired techniques for visualizing time-varying data" VIS '05, Minneapolis, MN, pp. 679-686, 2005.

[21] P.O. Kristensson et al., "An evaluation of space time cube representation of spatiotemporal patterns," IEEE Trans. Vis. Comput. Graphics, vol. 15, no. 4, pp. 696-702, 2008.

[22] F. Amini, S. Rufiange, Z. Hossain, Q. Ventura, P. Irani, M.J. McGuffin, "The impact of interactivity on comprehending 2D and 3D visualizations of movement data," IEEE Trans. Vis. Comput. Graphics, vol. 21, no. 1, pp. 122-135, 2015.

[23] T. Weinkauf, H. Theisel, A.V. Geider, A. Pang, "Stable feature flow fields," IEEE Trans. Vis. Comput. Graphics, vol. 17, no. 6, pp. 770-780, 2011.

[24] G. K. Karch, F. Sadlo, D. Weiskopf, T. Ertl, "Visualization of 2D unsteady flow using streamline-based concepts in space-time," J. Visual., vol. 19, no. 1, pp. 115-128, 2016.

[25] F. Sauer, K.-L. Ma, "Spatio-temporal feature exploration in combined particle/volume reference frames," IEEE Trans. Vis. Comput. Graphics, vol. 23, no. 6, pp. 1624-1635, 2017.

[26] H.E. Fiedler, "Coherent structures in turbulent flows," Prog. Aerosp. Sci., vol. 25, no. 3, pp. 231-269, 1988.

[27] S. Ferrari et al., "Visualizing vortex clusters in the wake of a high-speed train," IEEE SMC 2017, Banff, Canada, pp. 683-688, 2017.

[28] B.P. Epps, "Review of vortex identification methods," 55th AIAA ASM. Texas, USA. 2017.

[29] T. Günther, H. Theisel, "The state of the art in vortex extraction," Comput. Graph. Forum, pp. 1-25, 2018.

[30] R.C. Strawn, D.N. Kenwright, J. Ahmad, "Computer visualization of vortex wake systems," AIAA Journal, vol. 37, no. 4, pp. 511-512, 1999.

[31] J. Jeong, F. Hussain, "On the identification of a vortex," J. Fluid Mech., vol. 285, pp. 69-94, 1995.

[32] J.C.R. Hunt, A.A. Wray, P. Moin, "Eddies, streams, and convergence zones in trubulent flows," Center for Trubulence Research Report, CTR-S88, pp. 193-208, 1988.

[33] S. Ferrari, Y. Hu, "Multi-domain correlation for vortex extraction in fluid flow fields," IEEE SMC 2015, Hong Kong, China. pp. 917-922, 2015.

[34] D. Eberly, R. Gardner, B. Morse, S. Pizer, C. Scharlach, "Ridges for image analysis," J. Math. Imaging Vis., vol. 4, no. 4, pp. 353-373, 1994.

[35] R. Peikert, M. Roth, "The "parallel vectors" operator – a vector field visualization primitive," VIS '99, San Francisco, CA., pp. 263-270, 1999.

[36] X. Tricoche, G. Kindlmann, C.-F. Westin, "Invariant crease lines for topological and structural analysis of tensor fields," IEEE Trans. Vis. Comput. Graphics, vol. 14, no. 6, pp. 1627-1634, 2008.

[37] R. Peikert, F. Sadlo, "Height ridge computation and filtering for visualization," IEEE Pac. Vis. Symp. '08, Kyoto, Japan, pp. 119-126, 2008.

[38] J. Sahner, T. Weinkauf, H.-C. Hege, "Galilean invariant extraction and iconic representation of vortex core lines," EuroVis '05, Leeds, UK., pp. 151-160, 2005.

[39] A.F. Frangi, W.J. Niessen, K.L. Vincken, M.A. Viergever, "Multiscale vessel enhancement filtering," MICCAI '98, Cambridge, MA., vol. 1496, pp. 130-137, 1998.

[40] J. Sahner, T. Weinkauf, N. Teuber, H.-C. Hege, "Vortex and strain skeletons in Eulerian and Lagrangian frames," IEEE Trans. Vis. Comput. Graphics, vol. 13, no. 5, pp. 980-990, 2007.

[41] T. Schafhitzel, et al., "Topology-preserving λ2-based vortex core line detection for flow visualization," Comput. Graph. Forum, vol. 27, no. 3, pp. 1023-1030, 2008.

[42] D.C. Banks, B.A. Singer, "A predictor-corrector technique for visualizing unsteady flow," IEEE Trans. Vis. Comput. Graphics, vol. 1, no. 2, pp. 151-163, 1995.

[43] M. Roth, R. Peikert, "A higher-order method for finding vortex core lines," VIS '98, Research Triangle Park, NC, pp. 143-150, 1998.

[44] J. Canny, "A computational approach to edge detection," IEEE Trans. Pattern Anal. Mach. Intell., vol. 8, no. 6, pp. 679-698, 1986.

[45] J. Clyne, P. Mininni, A. Norton, "Physically-based feature tracking for CFD data," IEEE Trans. Vis. Comput. Graphics, vol. 19, no. 6, pp. 1020-1033, 2013.

[46] T. Schafhitzel, K. Baysal, U. Rist, D. Weiskopf, T. Ertl, "Particle-based vortex core line tracking taking into account vortex dynamics," ISFV13 / FLUVISU12, Nice, France, pp. 1-12, 2008.

[47] F. Reinders, F.H. Post, J.W. Spoelder, "Visualization of time-dependent data using feature tracking," Visual Comput., vol. 17, no. 1, pp. 55-71, 2001.

[48] B. Vrolijk, F. Reinders, F.H. Post, "Feature tracking with skeleton graphs," Data Visualization, Boston, MA., Springer US, pp. 37-52, 2003.

[49] A. Dimitrov, M. Golparvar-Fard, "Robust NURBS surface fitting from unorganized 3D point clouds for infrastructure as-built modeling," ICCCBE 2014, Orlando, FL., pp. 81-88, 2014.

[50] C. de Boor, A practical guide to splines, revised ed., Springer-Verlag, 2003

[51] P. Walker, "Depicting visual motion in still images: forward leaning and a left to right bias for lateral movement," Perception, vol. 44, no. 2, pp. 111-128, 2015.

[52] M. Hummel, C. Garth, B. Hamann, H. Hagen, K.I. Joy, "IRIS: illustrative rendering for integral surfaces," IEEE Trans. Vis. Comput. Graphics, vol. 16, no. 6, pp. 1319-1328, 2010.

[53] Kitware, "VTK." [Online]. Available: https://www.vtk.org/

[54] Z. Hosseini, R.J. Martinuzzi, B.R. Noack, "Sensor based estimation of the velocity in the wake of a low aspect ratio pyramid," Exp. Fluids, vol. 56, no. 13, pp. 1-16, 2015.

[55] A. Hemmati, D.H. Wood, R.J. Martinuzzi, "Characteristics of distinct flow regimes in the wake of an infinite span normal thin flat plate," Intl. J. Heat Fluid Fl., vol. 62, part B, pp. 423-436, 2016.

[56] R. Rozenholtz, Y. Li, L. Nakano, "Measuring visual clutter," J. Vis, vol. 7, no. 2, pp. 1-22, 2007.

[57] J.M. Henderson, M. Chanceaux, T.J. Smith, "The influence of clutter on real-world scene search: evidence from search efficiency and eye movements," J. Vis., vol. 9, no. 1, pp. 1-8, 2009.

[58] J. Reininghaus, J. Kasten, T. Weinkauf, I. Hotz, "Efficient computation of combinatorial feature flow fields," IEEE Trans.Vis. Comput. Graphics, vol. 18, no. 9, pp. 1563-1573, 2011.