

Exploring 3D Unsteady Flow using 6D Observer Space Interactions

Xingdi Zhang , Amani Ageeli , Thomas Theußl , Markus Hadwiger , and Peter Rautek 

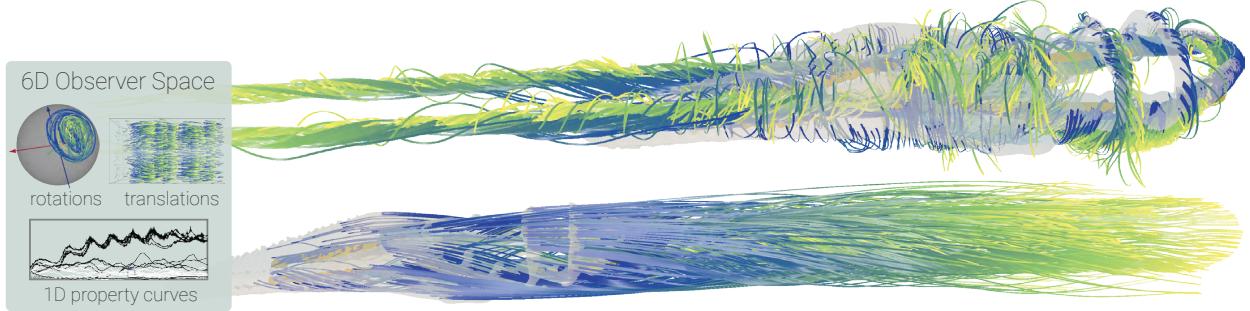


Fig. 1: Flow phenomena in a Delta Wing simulation, observed in co-moving vs. lab frame. (Top row) Snapshot in *co-moving frame*. We visualize a semi-transparent isosurface of vorticity magnitude together with pathlines seeded near regions of high vorticity. With our observer space framework, we can identify individual reference frames that co-move with certain features in the flow. (Bottom row) The same isosurface and pathlines in the *lab frame*. In this frame, the pathlines advect collectively downstream (to the right), reflecting the global translation of the flow. Thus, pathlines in the co-moving frame reveal the swirling behavior better than in the lab frame.

Abstract—Visualizing and analyzing 3D unsteady flow fields is a very challenging task. We approach this problem by leveraging the mathematical foundations of 3D observer fields to explore and analyze 3D flows in reference frames that are more suitable to visual analysis than the input reference frame. We design novel interactive tools for determining, filtering, and combining reference frames for observer-aware 3D unsteady flow visualization. We represent the space of reference frame motions in a 3D spatial domain via a 6D parameter space, in which every observer is a time-dependent curve. Our framework supports operations in this 6D observer space by separately focusing on two 3D subspaces, for 3D translations, and 3D rotations, respectively. We show that this approach facilitates a variety of interactions with 3D flow fields. Building on the interactive selection of observers, we furthermore introduce novel techniques such as observer-aware streamline- and pathline-filtering as well as observer-aware isosurface animations of scalar fluid properties for the enhanced visualization and analysis of 3D unsteady flows. We discuss the theoretical underpinnings as well as practical implementation considerations of our approach, and demonstrate the benefits of its 6+1D observer-based methodology on several 3D unsteady flow datasets.

Index Terms—Flow visualization, unsteady flow, reference frame optimization, interactive visualization, coherent structures

1 INTRODUCTION

The study of vortices and coherent structures is a cornerstone of fluid dynamics, providing essential insights into the complexities of unsteady flow fields. These patterns of fluid motion play a critical role in a wide array of natural and technological phenomena, ranging from aerodynamics to the mixing processes in the atmosphere and oceans, and even the propulsion mechanisms of biological organisms. The ability to accurately model these structures is therefore of paramount importance across various scientific and engineering disciplines.

While two-dimensional models have proven useful for understanding certain aspects of fluid flow, they inherently neglect the intricate dynamics that arise in three dimensions. Many significant flow phenomena, including vortex rings and helical vortices, possess an inherently three-dimensional nature that cannot be adequately captured in 2D. Phenomena that exhibit intricate fluid structures make it notoriously difficult to choose effective visual representations for visualization.

Common approaches include streamlines, pathlines, isosurfaces, and the extraction and visualization of topological structures such as vortex core lines [8, 20]. Because of the complexity of the intricate fluid motions, the resulting visualizations can become cluttered and ambiguous, hampering the analyst’s ability to interpret subtle flow phenomena. In this paper, we address the need for three-dimensional analytical tools that help in understanding such complex flow structures.

A powerful countermeasure to this complexity is to exploit the idea of observer-aware visualization, where the user can directly interact with transformed versions of the 3D flow field by interactively changing the current reference frame. Because motion is inherently relative, these transformations do not alter the physics, but they can dramatically alter how different aspects of the flow are revealed. In many cases, moving to the right reference frame makes coherent structures, such as vortex cores, appear more stationary, thus simplifying the visual representation as well as analysis. However, the key question is which reference frame to choose. Even if we only consider physically realizable (i.e., global rigid-body) observers, the space of all possible rigid motions in three dimensions spans six degrees of freedom plus time, i.e., a vast 6+1D parameter space. Navigating this space effectively and finding frames that highlight specific features remain open challenges.

In this paper, we address these challenges with a set of guiding principles and interactive tools for working with 6D observer space. First, we emphasize interpretability: Subspaces that are straightforward to navigate and directly tied to physical intuition, such as 3D translations and 3D rotations. Second, we aim for a good feature fit: Co-moving with flow elements of interest, such as a particular vortex, so that these features become “as steady as possible” relative to the chosen observer.

• Xingdi Zhang, Amani Ageeli, Markus Hadwiger, and Peter Rautek are with King Abdullah University of Science and Technology (KAUST), Thuwal, 23955-6900, Saudi Arabia. E-mail: {xingdi.zhang | amani.ageeli | markus.hadwiger | peter.rautek}@kaust.edu.sa

• Thomas Theußl is with Consivi KG, Austria. Email: thomas.theussl@consivi.com

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

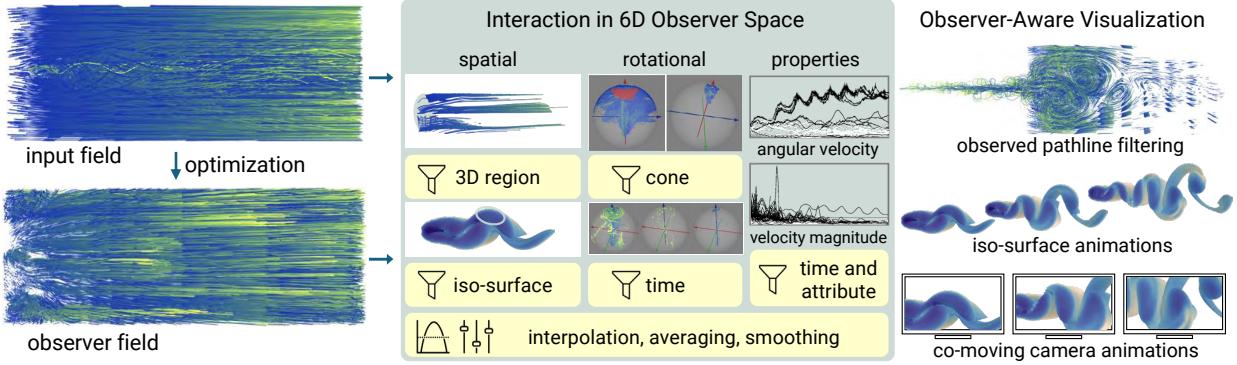


Fig. 2: Workflow overview. Our framework for 3D observer-aware visualization takes a time-dependent 3D vector field as input. We use optimization methods to attempt the extraction of observer fields that contain reference frames that co-move with features in the input flow. In our interaction framework, we split the 6D parameter space of reference frame transformations into a 3D translational and a 3D rotational subspace, respectively. Additionally, we compute properties of the reference frames including velocity magnitude and angular velocity. Each time-dependent reference frame transformation corresponds to one curve in each of these visualizations. The user can filter, combine, and smooth reference frame candidates using these plots, driving an interactive feedback loop that visualizes the input data with several observer-aware visualization methods.

Finally, we allow for smoothing to avoid abrupt, distracting changes in the reference frame’s motion. Building on these principles, we propose an interactive visualization framework that enables users to derive, inspect, and refine candidate reference frames, i.e., specific time-dependent curves in 6D observer space, and then apply them in *observer-aware* flow visualization and analysis.

Fig. 2 illustrates this workflow. On the left side, we depict automated methods that extract entire fields of potentially useful observer motions [7, 10]. The user can choose any pathline in an *observer field* to serve as the world line [10] of a suitable 3D observer motion. Additionally, we allow users to directly trace world lines in the input flow field. By tracing pathlines from user-defined seeds or neighborhoods, we create local reference frames co-moving with those trajectories.

In the center of Fig. 2, we show methods for interactively modifying, combining, and filtering the candidate reference frames with our framework. Choosing a single curve or merging multiple candidates is made straightforward by our framework, and is guided by the user’s current task, such as focusing on one specific vortex.

Finally, Fig. 2 (right) highlights three visualization methods enabled by reference frame-awareness. First, stream- and path-lines shown relative to a moving reference frame remove unnecessary global motion; swirling trajectories thus appear simpler. We propose an *interactive streamline- and pathline-filtering* strategy computing flow “steadiness” within the moving reference frame. Visualizing lines that become “almost” steady in the current reference frame can reveal coherent structures of interest in the flow field. While streamlines represent instantaneous phenomena, pathlines also provide temporal context. Second, we employ *observer-aware isosurface animation*. By co-moving with a target isosurface, we reduce aliasing and artifacts: We temporally up-sample the scalar field in the co-moving frame and visualize the isosurface animation as seen from a co-moving camera. Together, these elements (derivation and selection of candidate observers, interactive observer refinement, and observer-aware visualization) form a coherent and novel pipeline for interactive observer-aware 3D unsteady flow exploration and analysis. Leveraging suitable subspaces of the 6D observer space reveals important yet otherwise obscured fluid structures, supporting exploration with physically correct and interpretable visualizations.

This paper is organized as follows. Sec. 2 reviews prior work. Sec. 3 introduces the mathematical framework for physically realizable frame transformations. Sec. 4 outlines automatic and interactive methods for extracting reference frame from input vector fields. Sec. 5 discusses interactive observer visualization techniques for selecting, filtering, and refining candidate observers. Sec. 6 presents observer-aware visualization methods. Sec. 7 demonstrates results and advantages of our observer-based framework on various 3D unsteady fluid datasets.

2 RELATED WORK

Flow visualization in time-dependent settings inherently depends on the notion of an *observer*, i.e., a *reference frame* relative to which a velocity field is measured [10, 33, 36]. Different observers can perceive the same unsteady flow field in ways that substantially impact the extraction and visualization of characteristic features such as material boundaries or vortices. Early approaches in continuum mechanics emphasized the *objectivity* (frame-indifference) of physical quantities, ensuring that essential properties such as material surfaces or vortex boundaries remain consistent under arbitrary rigid motions [10, 15, 21, 33, 38]. While Galilean invariance addresses pure translations [4, 26], objectivity generalizes this to all rigid-body transformations [7, 13, 33].

Flow Feature Extraction. A large class of work has focused on detecting and analyzing coherent features such as vortex cores or separating material boundaries in time-dependent flows [5, 16, 17, 30, 34]. Region-based methods commonly rely on scalar invariants like the Q -criterion, the λ_2 -criterion, or the Okubo–Weiss approach [16, 17, 34], while line-based methods identify feature cores by tracking critical points or swirling streamlines [22, 30, 32]. In unsteady flow settings, Lagrangian perspectives become more appropriate to capture the motion of material surfaces [13, 27]. This includes finite-time Lyapunov exponent (FTLE) fields [12, 19, 27], and Lagrangian-averaged vorticity deviation (LAVD) [14], each providing objective characterizations of coherent structures. Ebling et al. [6] analyzed the situation where the flow is a superposition of different components (e.g., background flow and local vortical motion) and showed that simply adding a constant background flow can create or annihilate critical points, fundamentally altering the resulting topology. Wiebel et al. [35] introduced localized flow analysis, in which a divergence- and curl-free potential flow is subtracted from the velocity to reveal the region-specific component of the motion in both steady and time-dependent CFD data. Our observer-space formulation generalizes this idea: rather than canceling only boundary-induced laminar components, we remove (6+1D) inertial observer motion.

Reference Frames and Objectivity. Bhatia et al. [3] introduced internal reference frames that are extracted directly from the data by subtracting a space- and time-varying harmonic background flow. Removing this component yields a frame in which formerly unsteady vector fields become nearly steady improving flow analysis methods. Recent research has established that many feature-extraction techniques can be *objectivized* by explicitly formulating or transforming them to *objectively determined* moving reference frames [7, 10, 31]. Global optimizations for finding reference frames that minimize unsteadiness have been proposed, enabling domain-wide observer velocity fields [10, 24]. Locally optimal observers, on the other hand, adapt the transformation to the flow only in a small spatial or spatio-temporal neighborhood [7, 25]. Such approaches facilitate the visualization and detection of features (e.g., vortex cores) in *observed flow fields* that

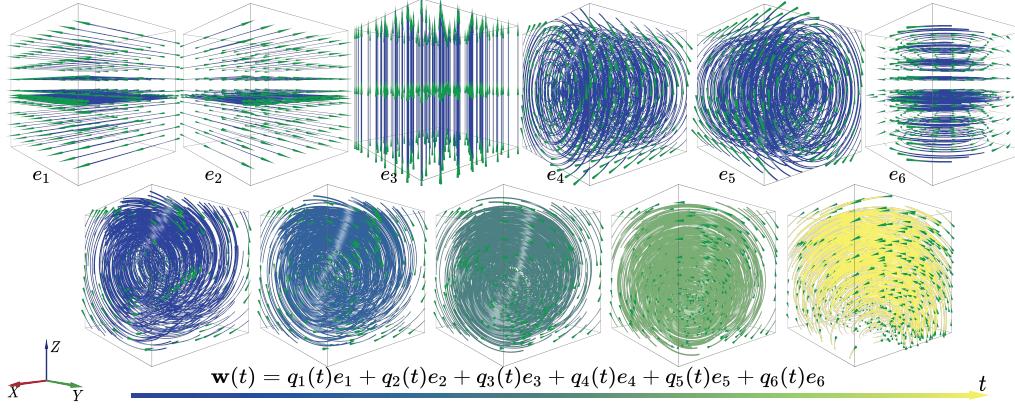


Fig. 3: (Top row) Six linearly-independent Killing vector fields $\{e_1, \dots, e_6\}$ form a basis of the vector space of all Killing vector fields in \mathbb{R}^3 . (Bottom row) Any time-dependent rigid motion $w(t)$ in \mathbb{R}^3 can therefore be expressed as a linear combination of these basis fields using six time-dependent coefficients $(q_1, \dots, q_6)(t)$, i.e., any $w(t) = \sum_i q_i(t) e_i$. In our framework, we visualize the parameters $q_i(t)$ in two separate sets $\{q_1, q_2, q_3\}$ (for all “translational” Killing fields) and $\{q_4, q_5, q_6\}$ (for all “rotational” Killing fields), respectively, each forming a 3D subspace of the space of all Killing fields.

locally become steady, or at least less unsteady, as seen by the optimally co-moving observer. Recent works also focus on the property that objective measures should be *physically observable* [18] or realizable, which our entire framework also fulfills.

Killing Vector Fields. A key mathematical tool for physically realizable observers is the notion of Killing vector fields, as the infinitesimal generators of rigid motions [23]. Approximate Killing fields have been employed in geometry processing and computer graphics (e.g., [2, 28]), and more recently in fluid flow exploration to define *observer fields* [10, 24]. Because they give isometric transformations, these fields ensure that fundamental physical properties, such as material boundaries and vortex shape, remain consistent for any observer [36].

Observer-based Visualization and Interaction Methods. Observer-based visualization methods have been proposed for vortex cores [9] in 2D and 3D fluid flow. Local observer-based techniques have been used in interactive systems [25], which enables real-time selection and visualization of locally optimal reference frames in user-defined lens regions. Similarly, observer extraction can be performed interactively from an optimally computed observer velocity field [36], letting users adapt the viewpoint to specific flow features on the fly. Such interactive reference frame transformations unify objective feature extraction with guided exploration, revealing coherent structures more clearly and enabling advanced tasks such as boundary tracking or local flow feature analysis [37].

Taken together, these methods demonstrate that observer fields and Killing vector fields can significantly aid feature detection, material boundary extraction, and visualization for unsteady flows. By leveraging objective transformations and local reference frames, modern techniques provide robust and physically meaningful insights into time-dependent flow phenomena. Prior work on *observer-relative* methods either focused on the *computation* of observers in 2D and 3D (e.g., [7, 10, 24]), on *visualization aspects* of 2D and 3D fluids (e.g., [9]), or *interaction methods* for 2D fluid flow only (e.g., [25, 36]). In contrast, in this paper, we address the additional challenges that arise from the complexity of 3D unsteady flow, with the first interactive 3D *observer-relative* visualization framework.

3 OBSERVERS IN \mathbb{R}^3

We give the mathematical background for observer-based visualization in three-dimensional Euclidean space \mathbb{R}^3 . We make use of time-dependent *Killing vector fields* in \mathbb{R}^3 , the SE(3) *Lie group* structure of rigid motions, and, most importantly, the $\mathfrak{se}(3)$ *Lie algebra* structure of infinitesimal rigid motions. Stillwell [29] provides an approachable introduction to Lie groups and Lie algebras, including an extensive discussion of their application to model rigid motions. A concise summary and a discussion of how Lie algebras can be used to model observer motions is given in the tutorial notes by Hadwiger et al. [11].

Although SE(3) is a *matrix Lie group*, and $\mathfrak{se}(3)$ the corresponding matrix Lie algebra, meaning that the elements of the group and algebra, respectively, are matrices, $\mathfrak{se}(3)$ is in one-to-one correspondence with (and even isomorphic to) the algebra¹ of all Killing vector fields on \mathbb{R}^3 . This space comprises all infinitesimal rigid motions in \mathbb{R}^3 .

Moreover, we define observer-relative quantities, such as observed velocity, observed velocity gradient, and observed time derivative.

3.1 Physically Realizable Observers in \mathbb{R}^3

A *physically realizable observer* in \mathbb{R}^3 is a reference frame that undergoes a time-dependent *rigid motion* relative to a fixed frame, e.g., the laboratory or lab frame. Rigid motions preserve distances and angles, and they are therefore exactly the transformations forming the *special Euclidean group* SE(3), which is the six-dimensional Lie group of translations and rotations in 3D Euclidean space.

Rigid Motions in \mathbb{R}^3 (Lie group SE(3), Lie algebra $\mathfrak{se}(3)$). The way in which any element of the special Euclidean group SE(3) acts on Euclidean space \mathbb{R}^3 can be expressed as a rigid transformation

$$\phi(y) = \mathbf{Q}y + \mathbf{b}, \quad y \in \mathbb{R}^3. \quad (1)$$

The 3×3 matrix $\mathbf{Q} \in \text{SO}(3)$ specifies a rotation in 3D, as an element of the special orthogonal Lie group SO(3), and $\mathbf{b} \in \mathbb{R}^3$ is a translation vector. In a time-dependent scenario, $\mathbf{Q}(t)$ and $\mathbf{b}(t)$ vary smoothly with time t . We can use a time-dependent rigid transformation $\phi_t(y)$ to describe a Euclidean *observer* or *reference frame* via the transformation from one frame (e.g., the lab frame) to another frame, as given by

$$x = \phi_t(y), \quad \text{with } \phi_t(y) = \mathbf{Q}(t)y + \mathbf{b}(t). \quad (2)$$

Time-Dependent Killing Fields in \mathbb{R}^3 . From a differential-geometric viewpoint, rigid motions in \mathbb{R}^3 are exactly those flows generated by *Killing vector fields* w , isomorphic to elements of $\mathfrak{se}(3)$. They satisfy

$$\nabla w(x)^T = -\nabla w(x).$$

That is, ∇w is skew-symmetric at every point $x \in \mathbb{R}^3$. A *time-dependent* Killing field in \mathbb{R}^3 is any $w(x, t)$ such that, for each fixed t , $w(x, t)$ is a Killing vector field. By taking the time derivative of Eq. 2, we obtain the corresponding *infinitesimal* rigid motion as the Killing field

$$w(x, t) = \Omega(t)x + \dot{\mathbf{b}}(t), \quad \text{with } \Omega(t) = \dot{\mathbf{Q}}\mathbf{Q}^T(t). \quad (3)$$

Here, the condition $\nabla w^T = -\nabla w$ is fulfilled, because $\nabla w = \Omega$, and

$$\Omega(t)^T = -\Omega(t). \quad (4)$$

¹An algebra is a vector space with an additional product operation. For the Lie algebra $\mathfrak{se}(3)$, this product is the matrix commutator $[X, Y] = XY - YX$, and in the case of Killing fields it is the Lie bracket $[X, Y]$ between vector fields X, Y .

The spin tensor $\Omega(t)$ corresponds to the angular velocity of the reference frame motion, i.e., to the derivative of its time-varying rotation, and the time-dependent vector $\dot{\mathbf{b}}(t)$ determines a time-varying velocity vector, i.e., the derivative of the time-dependent translation vector $\mathbf{b}(t)$.

All physically realizable observers in \mathbb{R}^3 , corresponding to all (infinitesimal) rigid motions in \mathbb{R}^3 , can be represented in this manner.

3.2 Lie Algebra of 3D Observer Motions

The set of all Killing fields in \mathbb{R}^3 forms a six-dimensional *Lie algebra*, commonly denoted $\mathfrak{se}(3)$. It comprises three translational degrees of freedom (the three components of \mathbf{b}) and three rotational degrees of freedom (the three degrees of freedom of the skew-symmetric Ω).

Basis Killing Fields in \mathbb{R}^3 . A basis for the vector space of all Killing fields in \mathbb{R}^3 , corresponding to the Lie algebra $\mathfrak{se}(3)$, is given by any linearly-independent set of Killing fields $\{\mathbf{e}_i\}$, $i \in \{1, \dots, 6\}$. We use

$$\mathbf{e}_1(x) = (1, 0, 0)^T, \quad \mathbf{e}_2(x) = (0, 1, 0)^T, \quad \mathbf{e}_3(x) = (0, 0, 1)^T, \quad (5)$$

$$\mathbf{e}_4(x) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} x, \quad \mathbf{e}_5(x) = \begin{pmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} x, \quad \mathbf{e}_6(x) = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} x.$$

Here, $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ represent translation velocities parallel to the x, y, z axes, respectively, while $\mathbf{e}_4, \mathbf{e}_5, \mathbf{e}_6$ represent infinitesimal rotations around the x, y, z axes, respectively. Fig. 3 depicts visualizations of these six basis Killing fields \mathbf{e}_i used throughout this paper.

Time-Dependent Observer Representation. Any time-dependent Killing field $\mathbf{w}(x, t)$ can be written in terms of these basis fields, where

$$\mathbf{w}(x, t) = \sum_{i=1..6} q_i(t) \mathbf{e}_i(x). \quad (6)$$

The $q_i(t)$ are six real coefficient functions that capture how translation and angular velocities, respectively, of the observer evolve over time. Using these coefficient functions $q_i(t)$ makes it straightforward to (1) *Visualize* time-dependent reference frame transformations as curves in different meaningful parameter spaces, (2) *Interpolate or average* multiple observers simply by interpolating or averaging their coefficients $q_i(t)$, and (3) *Compute differences* between observers by defining an inner product on the space of Killing fields corresponding to $\mathfrak{se}(3)$.

3.3 Reference Frame Transformations in 3D

Given the lab frame (or really any reference frame, but for easier understanding we will use the term lab frame here), and a description of *another* (e.g., observer) reference frame via a Killing field $\mathbf{w}(x, t)$, we can compute the time-dependent rigid (i.e., distance-preserving) map ϕ_t transforming any point into the new frame. Corresponding to Eq. 2, we compute the function $\mathbf{b}(t) \in \mathbb{R}^3$ such that

$$\dot{\mathbf{b}}(t) = \mathbf{w}(\mathbf{b}(t), t), \quad (7)$$

where $\dot{\mathbf{b}}(t) = (d/dt)\mathbf{b}(t)$. We also compute $\mathbf{Q}(t) \in \text{SO}(3)$ such that

$$\dot{\mathbf{Q}}(t) = \nabla \mathbf{w}(t) \mathbf{Q}(t), \quad (8)$$

where $\nabla \mathbf{w} = \Omega$ is the skew-symmetric tensor encoding the instantaneous rotation of the new frame, as measured by the lab frame.

Given the map ϕ_t (Eq. 2), the corresponding *pullback* ϕ_t^* then transfers objective vector and tensor fields from the lab frame to the new observer's frame. In Euclidean space \mathbb{R}^3 , we simply have

$$\phi_t^* = \mathbf{Q}^T. \quad (9)$$

Objective vector fields \mathbf{s} , or tensor fields \mathbf{S} , transform, respectively, as

$$\mathbf{s}^* = \mathbf{Q}^T \mathbf{s}, \quad \mathbf{S}^* = \mathbf{Q}^T \mathbf{S} \mathbf{Q}. \quad (10)$$

3.4 Observer-Relative Quantities in \mathbb{R}^3

An important step is computing the flow field and its derivatives as seen by the observer. Let $\mathbf{v}(x, t)$ be an unsteady velocity field. Then we can define **Observed Velocity** as

$$\mathbf{v}^*(y, t) = \phi_t^*((\mathbf{v} - \mathbf{w})(x, t)). \quad (11)$$

Here, $\mathbf{v} - \mathbf{w}$ is the relative velocity of fluid parcels w.r.t. the observer, and $\phi_t^* = \mathbf{Q}^T$ accounts for the rotation between the two observers. The **Observed Velocity Gradient** is defined as

$$\nabla \mathbf{v}^*(y, t) = \phi_t^*((\nabla \mathbf{v} - \nabla \mathbf{w})(x, t)). \quad (12)$$

Because $\nabla \mathbf{w}$ is skew-symmetric, apart from rotation by \mathbf{Q}^T only the skew-symmetric part of $\nabla \mathbf{v}$ is affected by the frame transformation.

The **Observed Time Derivative** is defined as

$$\frac{\partial \mathbf{v}^*}{\partial t}(y, t) = \phi_t^*\left(\left(\frac{\partial \mathbf{v}}{\partial t} - \frac{\partial \mathbf{w}}{\partial t} + \nabla \mathbf{v} \cdot \mathbf{w} - \nabla \mathbf{w} \cdot \mathbf{v}\right)(x, t)\right). \quad (13)$$

This is the rate of change of the field \mathbf{v} as measured by the moving observer, in terms of quantities measured in the lab frame. Transformed into the observer's reference frame, it is simply the time partial of the observed velocity \mathbf{v}^* . Here, the terms $\nabla \mathbf{v} \cdot \mathbf{w} = \nabla \mathbf{w} \mathbf{v}$ denote the directional derivative of the vector field \mathbf{v} in direction \mathbf{w} , and vice versa for $\nabla \mathbf{w} \cdot \mathbf{v} = \nabla \mathbf{v} \mathbf{w}$, respectively.

4 DETERMINING OBSERVERS FOR 3D UNSTEADY FLOW

A fundamental property of our 3D framework is that we *determine new observers* in ways that are indifferent to the reference frame of the input data. As in our prior work on 2D flow [36], this implies that *properties of the input velocity field computed relative to any such 3D observer are objective*. Because we also restrict observers to be *physically realizable*, i.e., to rigid transformations in \mathbb{R}^3 , these properties are also physically meaningful and measurable in reality [18]. Interpolation or averaging of physically realizable observers as 3D Killing fields remains objective as well, preserving the same fundamental principles.

4.1 Determining Observers in 3D

From a fundamental property of Killing vector fields on smooth manifolds, any Killing field \mathbf{w} is uniquely identified by a pair $(\mathbf{w}, \nabla \mathbf{w})(p)$ at a given point p . For a time-dependent Killing field, this means that the field is fully specified by $(\mathbf{w}, \nabla \mathbf{w})(p(t), t)$, at a given point $p(t)$, for each time t . In our context, this means that we can uniquely specify an *observer* by a time-dependent function

$$t \mapsto (p(t), \mathbf{w}(p(t), t), (\nabla \mathbf{w})(p(t), t)). \quad (14)$$

We refer to the time-dependent curve $t \mapsto p(t) \in \mathbb{R}^3$ as a *world line* [10]. By definition, the world line $t \mapsto p(t)$ must be a solution of the ODE

$$\frac{d}{dt} p(t) = \mathbf{w}(p(t), t), \quad p(0) = p_0. \quad (15)$$

By knowing one vector $\mathbf{w}(p(t), t)$ and one 3×3 skew-symmetric matrix $(\nabla \mathbf{w})(p(t), t)$ (with only three degrees of freedom) for each point $p(t)$ along $t \mapsto p(t)$, we can reconstruct the entire 3D Killing field \mathbf{w} by

$$\mathbf{w}(x, t) = \mathbf{w}(p(t), t) + (\nabla \mathbf{w})(p(t), t) \cdot (x - p(t)). \quad (16)$$

For more details see Appendix D.

4.2 Extracting Observers in 3D Euclidean Space

We can *extract* any number of 3D observers from a given 3D observer velocity field $\mathbf{u}(x, t)$ [10], which is, in general, *not* a Killing field. To do this, we define a world line $t \mapsto p(t)$ to be a *pathline* of the field \mathbf{u} . That is, from an initial position p_0 , we solve the ODE

$$\frac{d}{dt} p(t) = \mathbf{u}(p(t), t), \quad p(0) = p_0. \quad (17)$$

The observer field \mathbf{u} itself may either be set to the input velocity field \mathbf{v} , i.e., $\mathbf{u} := \mathbf{v}$, or result from another objective procedure or optimization [10, 24, 36]. Each world line obtained in this way is Lagrangian, and thus objective [14]. Thus, the corresponding observer is obtained in a reference frame-indifferent way. Additionally, any observer world line will provide us with a physically realizable rigid motion observer.

Given a 3D observer field $\mathbf{u}(x, t)$, we can obtain observers by

1. Picking an initial point p_0 at any $t = t_0$,
2. Solving the path-line ODE (Eq. 17) from p_0 to obtain $t \mapsto p(t)$,
3. Determining the observer spin tensor $\nabla \mathbf{w}$ at each point $p(t)$.

We determine the spin tensor $\nabla \mathbf{w}$ to identically match the *skewsymmetric part* of $\nabla \mathbf{u}$ at the point $(p(t), t)$. That is, we define

$$(\nabla \mathbf{w})(p(t), t) := \frac{1}{2} \left((\nabla \mathbf{u}) - (\nabla \mathbf{u})^T \right)(p(t), t). \quad (18)$$

5 REPRESENTATION, VISUALIZATION, AND INTERACTION IN 6D OBSERVER SPACE

In our framework, observers for a 3D flow domain correspond to curves $t \mapsto (q_i)(t)$ ($i = 1, \dots, 6$) in a six-dimensional parameter space \mathbb{R}^6 , with three parameters encoding the infinitesimal translation, and three parameters encoding the infinitesimal rotation, at each time t .

5.1 Representation in Two 3D Parameter Subspaces

By splitting each observer's coefficient vector $(q_i)(t)$ into (q_1, q_2, q_3) and (q_4, q_5, q_6) , we obtain two 3-dimensional parameter subspaces:

$$\underbrace{(q_1(t), q_2(t), q_3(t))}_{\text{infinitesimal translation}} \quad \text{and} \quad \underbrace{(q_4(t), q_5(t), q_6(t))}_{\text{infinitesimal rotation}}.$$

Each observer thus traces out a *curve* in each of these 3D parameter subspaces. Multiple observers then correspond to bundles or families of curves. Sets of observers can be visualized, explored, and selected by examining their projected paths in these two subspaces.

Translational subspace. For each observer, the infinitesimal translation parameters (q_1, q_2, q_3) correspond to the velocity of a reference point or *observer world lines* through space. Visualizing this subspace as \mathbb{R}^3 allows us to depict world lines as parametric trajectories relative to the lab frame. We show these curves as paths in (q_1, q_2, q_3) -space, color-coded or annotated by time, effectively acting like *3D path plots of the translational velocity*.

Rotational subspace. Similarly, (q_4, q_5, q_6) describe the infinitesimal rotation, i.e., the observer's angular velocity, i.e., axis and angular speed of rotation. Since the angular velocity magnitude can become large, we optionally normalize or clamp it for visualization. For $\|(q_4, q_5, q_6)\| > 1$, we clamp the vector to the unit sphere S^2 , so that the *direction* becomes more visually salient, whereas the magnitude can be color-coded or given by annotations. Thus, we obtain a *direction sphere* (Fig. 1, top left) depicting instantaneous axes of observer rotation.

1D Property Plots. Beyond storing (q_1, \dots, q_6) over time, we also track derived scalar properties such as the *Translational Speed* ($\|(q_1, q_2, q_3)\|$), the *Rotational Speed* ($\|(q_4, q_5, q_6)\|$), the *Magnitude of the Observed Time Derivative* (see Eq. 13), and other physical measures (e.g., strain rate) in the co-moving frame. Plotting these scalar functions of time yields 1D graphs that provide additional insight: Large spikes or plateaus may correspond to dynamically distinct motions of the observer. Moreover, the same property plots enable the user to *filter* subsets of observers based on thresholds in these derived properties (see Sec. 5.2).

Visual Interaction in Subspaces. Because each observer is now a parametric curve in two separate 3D spaces (plus optional 1D property plots), we allow interactive operations such as clicking or brushing in the translational subspace to select (or highlight) a set of observers, or specifying an angular wedge or region on the sphere to filter out certain rotations in the rotational subspace. Combined with time-based selection in the property plots, these subspace representations serve as visual handles for choosing or modifying observers, merging or splitting sets of observers, and revealing key events.

5.2 Selection and Filtering in Observer Subspaces

Once sets of time-dependent observers are visualized as curves in the mentioned parameter spaces, we can apply a variety of filters to declutter the view or to extract sub-families of interest.

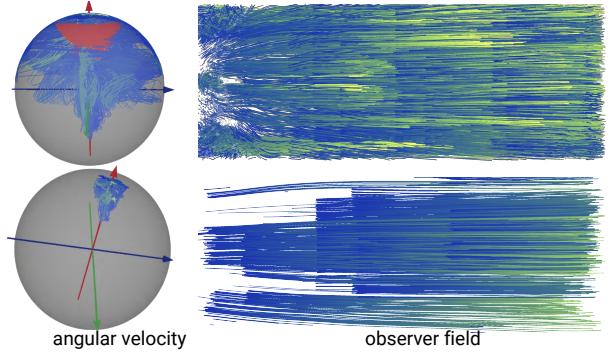


Fig. 4: **Filtering in rotational subspace.** (Top row) shows the observer field obtained via numerical optimization (right), along with its corresponding curves in the rotational subspace (left). A cone filter (red cone geometry) is applied to remove world lines whose angular velocities fall outside the desired range. (Bottom row) displays the filtered world lines (right) and their corresponding curves in the rotational subspace (left).

Filtering by Spatial Region. Spatial filtering methods include, but are not limited to, proximity to isosurfaces and dynamic bounding boxes traced from pathline clusters (see Appendix A). In the observer spatial subspace, because (q_1, q_2, q_3) typically encodes rigid-body translations or observer reference points, restricting these coefficients to a user-specified region filters out observers not translating through that area with the desired velocity.

Rotation-Axis Filtering. In the rotational subspace (q_4, q_5, q_6) , we can treat the unit sphere S^2 as the domain of all rotation axes, enabling interactive *conical* or *angular* selections. Concretely, we let the user choose a cone around a preferred axis, e.g., restricting to all observers whose rotation axis stays close to a major axis of rotation of a vortex street. This is especially helpful when many observer curves overlap or cluster in the rotational space. Fig. 4 shows an example of using cone filtering to constrain the observer's rotational subspace.

Time-Window Pre-Filtering. Some observers' trajectories in these subspaces become complicated over a long time interval. To reduce visual clutter, we allow restricting the view to a short time window, e.g., $[t_{\text{start}}, t_{\text{end}}]$, effectively hiding all observer curves outside it. This not only clarifies how the curves are distributed in the subspace over that interval, but also enables interactive *time animation* by sliding t_{start} or t_{end} to reveal how observers enter or leave a region. Furthermore, such a time-window filter aids in identifying clusters of "co-moving" frames that persist for only part of the simulation.

1D Property-Based Selection. Finally, we link subspace views to property plots showing scalar observer metrics (e.g., translational speed, rotation magnitude, streamline pathline similarity, etc.). Thresholds on $\|(q_1, q_2, q_3)\|$ hide observers whose translational velocities exceed desired ranges; similarly, thresholds on $\|(q_4, q_5, q_6)\|$ distinguish slowly rotating observers from fast-spinning ones. Users brush scalar curve ranges to highlight or exclude observers exceeding selected thresholds. This filtering allows users to refine the set of observers to a subset meeting specific physical constraints (e.g., moderate rotation, low acceleration).

Operations on Sets of Observers With our framework users can make computations on sets of observers, such as interpolating between them or averaging them. Interpolation of observers allows us to render smooth transitions when switching between observers for all observer-aware visualization methods. Averaging of observers allows us to compute a 'representative' observer from a set of observers. For instance, if several co-moving observers are found for a single feature of interest, the average observer is a representative that can be used for observer-aware visualizations of this feature.

To *average* multiple observers $\{\mathbf{w}_k\}$ we simply average their coefficients at each time. Likewise, to *interpolate* between two observers $\mathbf{w}_1, \mathbf{w}_2$, with their time dependent coefficients $q_{1,i}(t)$ and $q_{2,i}(t)$ we do a convex combination

$$q_i(t; \lambda) = (1 - \lambda) q_{1,i}(t) + \lambda q_{2,i}(t), \quad 0 \leq \lambda \leq 1. \quad (19)$$

Another useful operation that is well defined on the space of observers is the inner product, which admits computation of differences or similarities of observers. We give a definition of the inner product and the associated distance between observers in Appendix C.

These operations, filters and selection techniques enable navigation through large sets of reference-frame transformations, focusing on those that follow or reveal features of interest. Observers not matching desired criteria (e.g., excessive spin or drift) are pruned, leaving a relevant subset for further analysis or final visualizations.

6 OBSERVER-AWARE FLOW VISUALIZATION METHODS IN 3D

In this section, we discuss *observer-aware* techniques for visualizing 3D unsteady flows. As outlined in previous work for 2D flow, these techniques rely on transforming a given fluid flow into the reference frame of a time-dependent observer. Such transformations can significantly reduce the apparent motion of flow features that simply advect through space, thereby focusing attention on intrinsic behavior such as deformation, swirling, or stretching. We extend these ideas to 3D fluid domains, including generalizations and extensions of existing 2D approaches. Specifically, we are revisiting and extending visualization techniques that include *observed pathlines*, *temporal interpolation of scalar fields*, and *co-moving cameras*.

In general, the design of observer-aware visualization algorithms is informed by the fact that there is not one single static view of the data. Instead, each distinct observer generates a *different* (physically interpretable) visualization. Although the underlying 3D flow is unchanged in its own continuum-mechanical sense, the *perceived* velocity, position relative to the camera, or isosurface motion can vary significantly from one observer to another. While this leads to a vast 6+1D space of possible observer-aware visualizations that might sound overwhelmingly complex, this plurality of valid visualizations can also be turned into an advantage: by comparing multiple reference-frame-aware images or animations, we can highlight essential flow phenomena that remain elusive in other frames. Using our framework, we let the user *navigate* the observer (and hence visualization) space, selecting or interpolating among candidate frames until the visualizations yield satisfactory answers.

Changes of Visual Representations. Observer transformations in 6+1D (3D translations, 3D rotations, 1D time) affect how visual representations change in three fundamentally different ways, that inform the implementation of the respective visualization algorithms:

(1) **Spatial primitives that experience rigid transformations:** Iso-surfaces, 3D volumetric renderings of scalar fields, or stream surfaces (in the fluid dynamical sense of a material boundary) correspond to purely spatial objects at each time slice. For scalar fields that do not change with relative velocity changes (due to observer motions), these visual representations merely undergo rigid transformations. Scalar fields with this property are called *invariant to reference frame transformations* or simply *objective*. Their *instantaneous shape* remains identical in the observed domain, which makes them good candidates to be precomputed and transformed afterwards with rigid body transformations. For instance, an evolving isosurface of an objective scalar value, can be made "locked in place" in the co-moving observer's reference frame. This leads to a clearer depiction of how the surface evolves in time.

(2) **Spatial primitives that experience general transformations:** Streamlines, or isosurfaces of non-objective scalar fields are also purely spatial objects, however since they are affected by the change of relative velocities, they completely change their shape and need to be recomputed for every observer or change of reference frame.

(3) **Lagrangian space-time primitives:** Pathlines, streaklines, and other *Lagrangian* curves or surfaces are inherently defined in a domain augmented by time. Transforming them from the lab frame to an observer's frame typically alters their shape and connectivity in 4D space-time. Hence, an observed pathline might look significantly more "steady" or swirl-like in the new reference frame, revealing coherent structures that were obscured in the lab frame. These primitives can be precomputed in the lab reference frame and are later transformed with a time-dependent rigid body transformation. While the precompu-

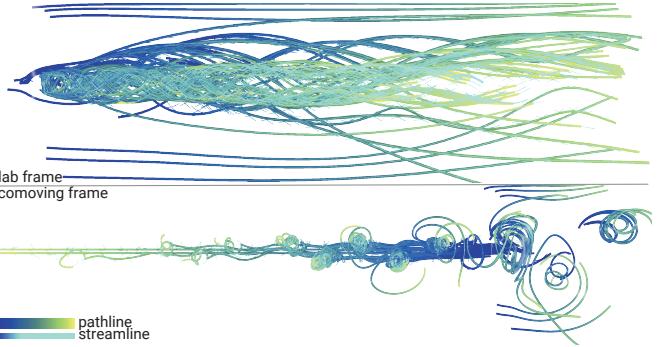


Fig. 5: Pathlines (time is color-coded) and streamline segments (light green) used to compute the steadiness of the flow in the current reference frame. (top) Lab frame. (Bottom) The same set of pathlines and streamline segments in the co-moving reference frame. In regions where the vector field becomes steady with respect to the current reference frame, pathlines become "streamline-like."

tation might involve computationally costly numerical integration or other extraction algorithms, the *observer-transformation* is typically computationally less costly and can be done in parallel.

Understanding the difference between these three classes helps to design efficient visualization algorithms. The subsections that follow, discuss specific examples.

6.1 Observed Streamline- and Pathline-Filtering

A core motivation for adopting time-dependent observers is to reveal regions of the flow that appear *nearly steady* in a particular *moving* reference frame. In practice, we want to find curves that represent parts of the fluid that look "stationary" when viewed in the observer's reference frame. This highlights regions that contain coherent structures of the fluid. However, trivial steady regions (where the flow field is steady in any reference frame) are typically uninteresting and shall also be filtered out. We thus propose a novel *two-stage filter* that (1) detects curve segments that represent locally near-steady flow regions in the current observer's reference frame; and (2) discards segments that are also near-steady in *any* frame, keeping only those that *gain* their steadiness from the current observer's perspective.

Before introducing our filter, we emphasize the complementary value of observer-relative streamlines and pathlines. Because streamlines are computed from a single time slice, transforming them into the moving reference frame yields an immediate, view of how the current observer choice reshapes the local flow geometry. In many scenarios, such as pinpointing an instantaneous vortex core or outlining shear interfaces, these observed streamlines convey the essential structure. In the Appendix, Fig. 13, we demonstrate this effect for the Boeing data set: when viewed in the co-moving frame, the streamlines collapse into tight spirals that delineate the vortex core far more cleanly than in the lab frame, underscoring their utility for frame-aware exploration. To extend this instantaneous visualization method, we propose to compute and filter from a large set of pathlines and show the ones that are locally similar to the streamlines. We propose the following method:

(1) **Detecting Near-Steady Regions in the Observer Frame.** We first measure how closely a pathline follows a streamline in the *observer's reference frame*. Concretely, we start with a pathline $\gamma(t)$ computed in the lab frame, sampled at discrete vertices $\gamma(t_k)$ (Sec. 3). At each pathline vertex $\gamma(t_k)$ and time t_k , we integrate a *local streamline* of the observed velocity field $\mathbf{v}^*(\mathbf{x}, t)$ (i.e. the velocity in the co-moving coordinates). This streamline integration is performed forward and backward for a short time τ , to produce a local streamline segment $S_k(t_k - \tau : t_k + \tau)$. We compute a distance

$$d_k := \frac{\int_{-\tau}^{\tau} (\gamma(t_k + s) - S_k(t_k + s))^2 ds}{2\tau} \quad (20)$$

that measures how far the pathline segment around $\gamma(t_k)$ deviates from the corresponding observer-frame streamline S_k . If d_k is below a threshold ϵ_{obs} , we classify that neighborhood of the pathline as *near-steady*

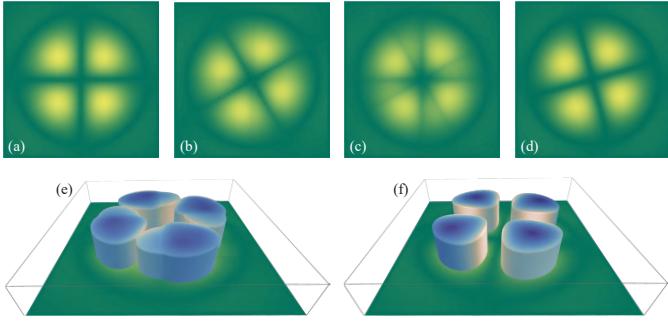


Fig. 6: Reference frame-aware isosurface extraction. (a,b) 2D slice of a 3D scalar field at time (a) t_0 , (b) t_1 . (c) Linear interpolation between (a), (b) at time $\frac{t_0+t_1}{2}$. (d) Reference frame-aware interpolation between (a), (b) at time $\frac{t_0+t_1}{2}$. (e) 3D iso surface extracted with linear interpolation at time $\frac{t_0+t_1}{2}$. (f) 3D iso surface extracted by first transforming and interpolating the scalar field in the co-moving reference frame at time $\frac{t_0+t_1}{2}$.

in the observer frame (i.e. “streamline-like”). In Fig. 5 we illustrate how the transformation of a set of pathlines and their corresponding streamline segments, into the reference frame of the co-moving observer, changes their geometry and alignment. Observed pathlines that are similar to streamlines in the co-moving frame of reference are of interest.

(2) Excluding Regions Steady in Any Frame. Not all near-steady regions are interesting—some might be stationary in any frame. We exclude trivial cases by computing the time derivative in the lab frame and discarding pathline segments in zero time-derivative regions. Hence, our final *observed pathline filter* displays only those segments that are streamline-like in the reference frame of the co-moving observer, but exhibit substantial unsteadiness in *any other* (e.g., the lab-) frame. The remaining trajectories highlight material parcels that gain their apparent steadiness only from the chosen observer’s viewpoint. By focusing on pathline segments that become steady *solely* under the current observer, we isolate Lagrangian coherent structures “co-moving” with the observer, such as vortex cores or rotating fluid parcels. Segments that remain near-stationary in all frames are discarded, thus removing steady fluid regions. When purely instantaneous insight is desired, streamlines can be filtered instead of pathlines with the identical two-stage filter, giving users a consistent tool for either temporal or instantaneous exploration.

The thresholds (ϵ_{obs} , ϵ_{lab}) can be tuned by the user to filter for features of interest. A smaller ϵ_{obs} retains strongly co-moving segments, while a larger ϵ_{lab} preserves contextual background flow regions. Because both thresholding and streamline integration can be done at interactive rates, domain scientists can explore different observer motions (e.g., solutions to global or local reference frame optimizations) and see which portions of the fluid remain steady, or swirl-like in each candidate observer frame. An example is shown in Fig. 8. This two-stage “streamline-like” matching strategy highlights flow regions most relevant to each observer’s perspective, avoiding clutter from globally trivial steady zones.

6.2 Scalar Field Interpolation for Isosurface Animations

Zhang et al. [37] demonstrated that extracting iso-lines or FTLE ridges in the co-moving reference frame leads to simpler shapes and more stable animations in 2D unsteady flow. Although this method generalizes to higher dimensions, the interpretation changes from iso-lines in 2D to *isosurfaces* or *FTLE surfaces* in 3D. Specifically, in 3D, surfaces are typically extracted at discrete timesteps as instantaneous spatial objects and then *animated* over time.

To generate intermediate frames, scalar values are usually interpolated in the lab frame, then isosurfaces are extracted. However, interpolated scalar fields often exhibit considerable artifacts from fast moving structures. By first transforming the scalar field into a suitable co-moving frame, we reduce artifacts that originate from the motion during interpolation. Hence, the resulting isosurfaces in intermediate frames appear more coherent.

Fig. 6 shows an example of an isosurface extracted in the lab frame compared to an isosurface extracted in the co-moving frame. The difference in Fig. 6 is large because we exaggerate the problem for the purpose of illustrating the effect of sub-optimal scalar field interpolation in the lab frame. To compute isosurface animations in co-moving frames, we choose an observer w whose motion is adapted to a region of interest (Sec. 3). At each timestep t_i , we *pull back* the scalar field $\varphi(x, t_i)$ into the observer’s coordinates via the rigid map $\phi_{t_i}^{-1}$ of w . This produces $\varphi^*(y, t_i) = \varphi(\phi_{t_i}(y), t_i)$, an observed scalar field in the moving frame. Because w co-moves with key features of interest, φ^* often varies less over time. Next, we interpolate this time series $\{\varphi^*(\cdot, t_i)\}$ at intermediate times (e.g. $t_{i+\alpha}$ with $0 < \alpha < 1$) to obtain a smooth animation in the observer coordinates. We extract isosurfaces in the observer’s reference frame and map the surfaces back to the lab frame. Formally:

1. Compute $\varphi^*(y, t_i) := \varphi(\phi_{t_i}(y), t_i)$ for all discrete t_i .
2. Interpolate $\varphi^*(y, t_{i+\alpha})$ over time in the co-moving domain.
3. Extract isosurfaces (or ridges) in $\varphi^*(y, t_{i+\alpha})$.
4. Map each surface back to the lab frame via $\phi_{t_{i+\alpha}}$ for visualization.

By performing interpolation in the co-moving frame, the scalar field often appears *less unsteady* for evolving structures that might primarily be undergoing translations or rotations. This reduces aliasing artifacts and smoothing errors during the animation of isosurfaces.

6.3 Co-moving Cameras via Local Reference Frames

A closely related yet distinct concept to scalar-field interpolation in a co-moving frame is to “lock” the camera onto a particular fluid feature. This allows the user to watch an evolving phenomenon (e.g., a vortex, an isosurface, or a specific sub-region of a larger structure) without the distraction of that feature drifting around in the scene. Prior 2D approaches [25, 37] demonstrated selecting an observer (rigid-body reference frame) to track and stabilize features. We generalize this idea to 3D flow visualization. In 3D, isosurfaces often span large regions with multiple loosely connected parts. The user might wish to “zoom in” on one localized phenomenon: e.g., a vortex within a larger structure. To address this, we let the user pick that sub-region directly on the isosurface at time t_0 , and compute an observer that specifically *tracks* or co-moves with that sub-region. Crucially, large connected surfaces may only partially move consistently (e.g., the middle of an isosurface might remain stationary, whereas its edges swirl). Being able to isolate a smaller region is key for effectively “locking onto” it.

Locking to a Targeted 3D Feature. In our framework, candidate observers are computed through user-driven region selection, extracting local reference frames by minimizing objectives like unsteadiness.

Local reference frames can be generated through two alternative approaches: applying a spatial filter after performing global observer optimization [7], or conducting spatial filtering first, followed by observer optimization within the localized area of interest. From a performance perspective, the latter approach generally proves more efficient and practical. In our framework, we define the local area of interest by integrating a cluster of pathlines and computing a time-dependent bounding box that encloses this cluster at each timestep. This results in a dynamically moving bounding box, as illustrated in Fig. 11 (Appendix A). If the feature is lost or moves out of view, users can adjust the observer accordingly. By repeating this process and smoothing the resulting observer, we obtain a time-dependent rigid transformation that keeps the chosen patch centered in the camera view.

Co-Moving Camera for Feature Animations. Once we obtained a suitable observer w , 3D visualizations (isosurface animations, volume rendering, pathlines) can be drawn in observer coordinates. From the user’s perspective, it appears as if the camera has locked onto the target sub-region: that region remains roughly stationary in the viewport while the rest of the flow (and bounding domain) may translate or rotate. Hence, *the user can focus on how the feature evolves* (e.g., *deforms*, *splits*, *merges*) over time, free from the complication of global drift, making it easier to perceive subtle morphological changes.

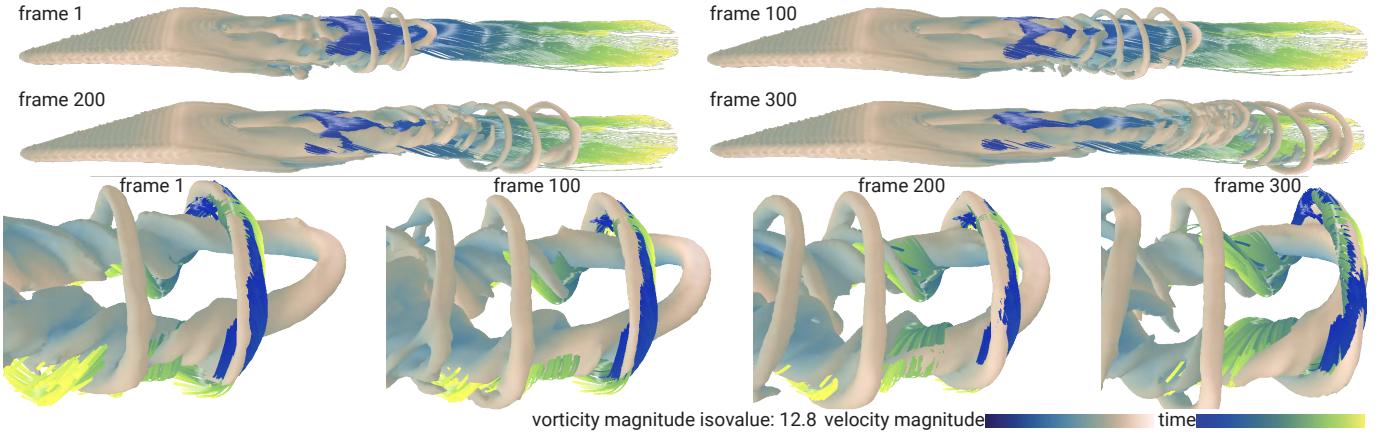


Fig. 7: **Animation sequence of an isosurface of vorticity magnitude.** (top) four frames of an animation sequence are shown in the lab-reference frame. (bottom) A locally adapted reference frame that co-moves with one specific vortex ring filament at the front, allows us to lock the camera to a small region of the isosurface. The user can focus on the evolution of in this region without the difficulty of following it while it drifts downstream.

7 RESULTS AND DISCUSSION

We evaluated our method on four different 3D unsteady flow datasets, each highlighting various aspects of our approach. In each case, we combined reference frame transformations, isosurface-based filtering, and pathline visualization to expose phenomena that are otherwise difficult to see in a standard lab-frame visualization. We also report performance numbers for the different computation-intensive components of our framework and created animation sequences (see video) to give a dynamic view on the presented case studies.

7.1 Case Studies

Delta Wing Simulation. In our first case study, we analyze vortex filaments behind a wing. The dataset is simulated using the Lattice Boltzmann Method with a grid resolution of $[110, 628, 110]$ over a spatial domain of $[-0.99, 1.00] \times [-5.71, 5.71] \times [-0.99, 1.00]$. The dataset spans a time range of $[0.00, 1.73]$ with 234 time steps. This dataset is characterized by multiple, intricately intertwined vortex filaments advecting downstream at different speeds. Traditional lab frame views make it hard to discern the swirling motion of each filament because they collectively translate in the flow direction. As shown in Fig. 1, we use an *isosurface-based spatial filter* (Sec. 5.2) to isolate regions of high vorticity, and seed pathlines near these vortex cores. These pathlines define the reference frame transformation described in Sec. 4.2. In the co-moving reference frame (Fig. 1, top row), the pathlines clearly swirl around the vortex filaments, while in contrast, the lab-frame visualization (bottom row) shows all pathlines convecting

downstream, largely obscuring the individual swirling characteristics of each filament. Fig. 7 focuses on a single vortex ring filament evolving at the leading edge. In the lab frame (top), this filament quickly drifts to the downstream region, forcing the viewer to follow it manually in a time-dependent animation. By contrast, using our isosurface animation and co-moving camera methods (Sects. 6.2 and 6.3), we are locking a locally adapted reference frame onto this filament (Fig. 7, bottom) and keep it within a fixed viewing volume. This makes changes of the shape and the relation to other filaments much more apparent. Because the Delta Wing flow includes multiple vortex filaments, no *single* reference frame can simultaneously “track” all filaments if they travel at different speeds or directions. Fig. 9 illustrates two filaments, each requiring

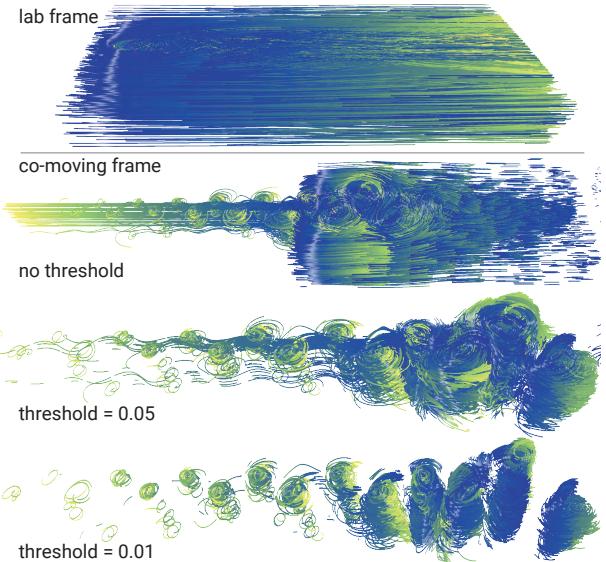


Fig. 8: **Pathline filtering** with different thresholds for the vortex street in the co-moving frame (bottom) compared to the lab frame (top).

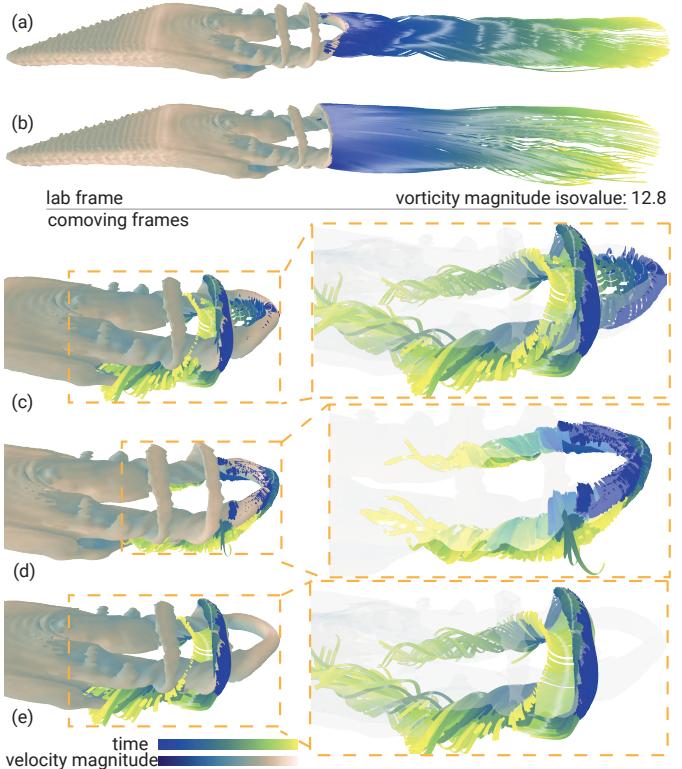


Fig. 9: **Observed pathlines** in different reference frames. Due to the different speeds of the vortex filaments we get two different co-moving frames. (a) and (b) two sets of pathlines, both in the lab frame. (bottom part) pathlines in different co-moving reference frames: (c) both sets of pathlines in the reference frame of the second vortex filament. One set of pathlines is not aligned with the vortex. (d) pathlines seeded on the first vortex filament in their co-moving reference frame. (e) pathlines seeded on the second vortex filament in their co-moving reference frame.

Table 1: Performance analysis of observed pathline filtering. The *Configuration* column specifies computational complexity parameters, giving: (number of pathlines, points per pathline, number of streamlines sampled per pathline, streamline integration steps). The *Observed Pathline* column is the CPU time required to transform pre-integrated pathlines into the observer's reference frame. The *Observed Streamline* column is the computational time needed to integrate observed streamlines at each sampled point along the pathlines. The last two rows show the timing of generating Fig. 8 and Fig. 10. All times are in ms.

| Configuration | Observed Pathline | | Observed Streamline | |
|-----------------------------------|-------------------|-------------|---------------------|-------------|
| | mean | range \pm | mean | range \pm |
| DeltaWing (64,640,64,100) | 0.82 | 0.20 | 295 | 37 |
| DeltaWing (256,2560,256,100) | 10.85 | 2.50 | 1450 | 200 |
| DeltaWing (1024,10240,1024,100) | 38.78 | 6.10 | 3500 | 300 |
| DeltaWing (1024,10240,1024,500) | 29.16 | 12.10 | 7530 | 900 |
| DeltaWing (4096,40960,4096,500) | 813.42 | 81.10 | 161280 | 5200 |
| Boeing (2048,2560,256,100) | 27.91 | 1.00 | 1720 | 200 |
| Half Cylinder (8196,1280,128,100) | 128.07 | 15.00 | 1050 | 200 |

its own co-moving frame. In Fig. 9(c), pathlines seeded for one filament appear *incorrectly* aligned when observed in the co-moving frame of another filament, demonstrating that each vortex core's swirling is frame-dependent. Conversely, Figs. 9(d) and 9(e) show each vortex filament in *its own* co-moving reference frame, revealing the clear swirling motion around each core. This discrepancy is not an artifact but evidence of how distinct vortex filaments demand individualized reference frames to capture their local dynamics accurately, which is a core motivation of our approach.

Half Cylinder Vortex Street is a 3D dataset capturing the flow behavior around a semi-circular obstruction on a structured grid [1]. The dataset has a spatial extent of $[-0.5, 7.5] \times [-1.5, 1.5] \times [-0.5, 0.5]$, a grid resolution of $[640, 240, 80]$, and a time domain of $[0, 15]$ with 151 timesteps. We computed reference frame transformations using local reference optimization [7]. We used our parameter space representation described in Section 5 to confirm that one reference frame transformation captures the evolving vortex street. In Fig. 8, we show results obtained with our pathline filtering method explained in Sec. 6.1.

Boeing 747 Simulation. This dataset was simulated using the Lattice Boltzmann Method, featuring a grid resolution of $[425, 950, 90]$ within a spatial domain of $[-2.49, 2.51] \times [-5.00, 5.02] \times [-0.99, 1.01]$. The dataset spans a time range of $[0.00, 1.96]$ with 199 time steps. To avoid costly computations of the reference frame transformations with optimization methods in regions of the dataset that are not of interest, our framework allows the user to select a subregion of interest on the isosurface (Sec. 5.2). We then compute the reference frame transformations using local reference optimization [7] only in the selected region and the advection of the selected region. This greatly improves the computation time and is more suitable in an interactive scenario. In Fig. 10, we show pathlines seeded in the lab-frame as well as the locally computed reference frame that was used to compute pathlines transformed into the co-moving reference frame. The pathlines were filtered with our pathline-streamline

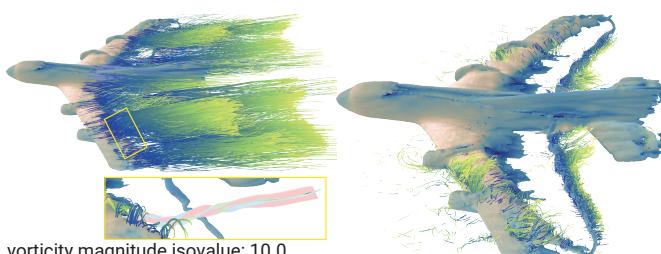


Fig. 10: Boeing 747 visualized using a moving reference frame extracted from a local optimization. (left) Pathlines are densely seeded on an isosurface of the Boeing 747 dataset. The yellow area marks the interaction area that the user selected to extract a reference frame. A local reference frame optimization is started in this area and the advected area over time, resulting in a reference frame that is visualized in the yellow rectangle below. (right) Pathlines are automatically filtered and only visualized in areas of the fluid flow that become nearly steady in the observed field.

Table 2: Time measurements for reference frame transformation of scalar field with 30 time steps (Rft-Scalar Field), marching cubes isosurface extraction for one time step (MC-Isosurface), and the generation of iso-surface animations (Animation) over 30 frames of different datasets.

| Dataset | Rft-Scalar Field | MC-Isosurface | Animation |
|---------------|---------------------|----------------------|--------------------|
| Boeing | 2.57 ± 0.50 sec | 678.10 ± 10.0 ms | 1.98 ± 0.2 sec |
| Half Cylinder | 2.98 ± 0.30 sec | 719.77 ± 26.8 ms | 1.65 ± 0.2 sec |
| DeltaWing | 3.99 ± 0.07 sec | 205.66 ± 33.8 ms | 1.12 ± 0.1 sec |

Table 3: Performance analysis of observer field optimization. *Global Time* is the runtime per timestep for global optimization [7]; note that this method fails to execute on the Boeing dataset due to memory consumption. *Local Optimizations* is the average number of grid points per timestep processed after local spatial filtering, and *Local Time* is the average runtime per timestep using our spatial filter. The first three rows correspond to Fig. 11 (a-c). The last row corresponds to Fig. 12.

| Dataset | Resolution x Timesteps | Global Time | Local Optimizations | Local Time |
|---------------|------------------------------|-------------|---------------------|------------|
| Boeing | $(425, 950, 90) \times 199$ | - | 5168 | 6.5 sec |
| Half Cylinder | $(640, 240, 80) \times 151$ | 420 sec | 1540 | 4.2 sec |
| DeltaWing | $(110, 628, 110) \times 234$ | 250 sec | 2925 | 6.8 sec |
| SmokeBuoyance | $(47, 95, 47) \times 160$ | 11 sec | 1100 | 2.0 sec |

similarity measure (Sec. 6.1) to highlight the regions of the fluid that become steady in the current frame of reference. Pathlines that align with the iso-surface indicate a coherent structure of particles that rotate in place (relative to the co-moving observer).

7.2 Performance

The performance evaluation experiments were conducted on an Intel(R) Xeon(R) Gold 6230R CPU @ 2.10GHz with 2 processors.

Performance of Observed Pathline Filtering. Pathlines are pre-integrated, and an interactive change of observer invokes two main computational steps: (1) transforming pathlines into the observer reference frame, and (2) integrating observed streamlines and computing the difference between them (Sec. 6.1). The computations are performed in parallel on the CPU. Timing measurements for these two major computational steps are reported in Table 1.

Co-moving Camera Isosurface animation. The co-moving camera animation involves two primary steps: (1) transforming the scalar field into the co-moving reference frame, and (2) extracting isosurfaces using the marching cubes algorithm at each animation frame. The generation of isosurface animation is done in parallel. The computational cost of isosurface extraction depends significantly on the selected iso-values. Table 2 reports the computational times corresponding to the representative iso-values and their resulting isosurfaces and isosurface animations presented throughout the paper. Pseudocode is given in Appendix B.

Local Reference Frame Optimization. In Table 3 we show performance measurements of the reference optimization method. We compare the time it takes to run the optimization of [7] on the entire grid for one time step (*Global Time*), with our method of first applying a spatial filter and then only locally extracting a reference frame transformation (*Local Time*). Although these times are not interactive, we significantly reduce computation times. For the Boeing dataset, we were not able to compute the reference frame optimization for the whole dataset on our desktop machine, but we were able to compute it for a spatially filtered region. Appendix A Fig. 11 shows the grid cells that were considered for the local optimization.

8 CONCLUSIONS

In this paper, we propose the first framework for interactive observer-aware visualization of 3D unsteady flow fields. Our framework is based on a unifying representation of reference frame motions as six time-dependent parameters. We use this representation to visualize, analyze, filter, and combine reference frame transformations in an interactive feedback loop. We presented novel, and adapted observer-aware visualization methods and applied them in different case studies. Our framework opens the door to more sophisticated 3D observer-aware analysis, which can further be integrated with topological and Lagrangian flow feature analysis methods in the future.

ACKNOWLEDGMENTS

This work was supported by King Abdullah University of Science and Technology (KAUST) baseline funding. This research used resources of the Core Labs of KAUST.

REFERENCES

- [1] I. Baeza Rojo and T. Günther. Vector field topology of time-dependent flows in a steady reference frame. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):280–290, 2020. doi: 10.1109/TVCG.2019.2934375 9
- [2] M. Ben-Chen, A. Butscher, J. Solomon, and L. Guibas. On discrete killing vector fields and patterns on surfaces. *Computer Graphics Forum*, 29(5):1701–1711, 2010. doi: 10.1111/j.1467-8659.2010.01779.x 3
- [3] H. Bhatia, V. Pascucci, R. M. Kirby, and P.-T. Bremer. Extracting features from time-dependent vector fields using internal reference frames. *Computer Graphics Forum*, 33(3):21–30, 2014. doi: 10.1111/cgf.12358 2
- [4] R. Bujack, M. Hlawitschka, and K. I. Joy. Topology-inspired Galilean invariant vector field analysis. In *Proceedings of IEEE Pacific Visualization 2016*, pp. 72–79, 2016. doi: 10.1109/PACIFICVIS.2016.7465253 2
- [5] R. Bujack, L. Yan, I. Hotz, C. Garth, and B. Wang. State of the art in time-dependent flow topology: Interpreting physical meaningfulness through mathematical properties. *Computer Graphics Forum*, 39(3):811–835, 2020. doi: 10.1111/cgf.14037 2
- [6] J. Ebling, A. Wiebel, C. Garth, and G. Scheuermann. Topology based flow analysis and superposition effects. In H. Hauser, H. Hagen, and H. Theisel, eds., *Topology-based Methods in Visualization*, pp. 91–103. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. doi: 10.1007/978-3-540-70823-0_7 2
- [7] T. Günther, M. Gross, and H. Theisel. Generic objective vortices for flow visualization. *ACM Transactions on Graphics*, 36(4):Article No. 141, 2017. doi: 10.1145/3072959.3073684 2, 3, 7, 9
- [8] T. Günther and H. Theisel. The state of the art in vortex extraction. *Computer Graphics Forum*, 37(6):149–173, 2018. doi: 10.1111/cgf.13319 1
- [9] T. Günther and H. Theisel. Objective lagrangian vortex cores and their visual representations. *IEEE Transactions on Visualization and Computer Graphics (IEEE Visualization)*, 31(1):76–85, 2025. doi: 10.1109/TVCG.2024.3456384 3
- [10] M. Hadwiger, M. Mlejnek, T. Theußl, and P. Rautek. Time-dependent flow seen through approximate observer Killing fields. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):1257–1266, 2019. doi: 10.1109/TVCG.2018.2864839 2, 3, 4
- [11] M. Hadwiger, T. Theußl, and P. Rautek. Riemannian geometry for scientific visualization. In *SIGGRAPH Asia 2022 Courses*, SA ’22, article no. 5, 82 pages. Association for Computing Machinery, New York, NY, USA, 2023. Latest version available at <https://vccvisualization.org/RiemannianGeometryTutorial/>. doi: 10.1145/3550495.3558227 3
- [12] G. Haller. Distinguished material surfaces and coherent structures in three-dimensional fluid flows. *Physica D: Nonlinear Phenomena*, 149:248–277, 03 2001. doi: 10.1016/S0167-2789(00)00199-8 2
- [13] G. Haller. An objective definition of a vortex. *Journal of Fluid Mechanics*, 525:1–26, 2005. doi: 10.1017/S0022112004002526 2
- [14] G. Haller, A. Hadjighasem, M. Farazmand, and F. Huhn. Defining coherent vortices objectively from the vorticity. *Journal of Fluid Mechanics*, 795:136–173, 2016. doi: 10.1017/jfm.2016.151 2, 4
- [15] G. A. Holzapfel. *Nonlinear Solid Mechanics: A Continuum Approach for Engineering*. Wiley, 2000. 2
- [16] J. C. R. Hunt, A. A. Wray, and P. Moin. Eddies, streams, and convergence zones in turbulent flows. In *Proceedings of the Summer Program 1988*, pp. 193–208. Center for Turbulence Research, Dec. 1988. 2
- [17] J. Jeong and F. Hussain. On the identification of a vortex. *Journal of Fluid Mechanics*, 285:69–94, 1995. doi: 10.1017/S0022112095000462 2
- [18] B. Kaszsás, T. Pedergnana, and G. Haller. The objective deformation component of a velocity field. *European Journal of Mechanics - B/Fluids*, 2022. doi: 10.1016/j.euromechflu.2022.12.007 3, 4
- [19] A. Kuhn, C. Rössl, T. Weinkauf, and H. Theisel. A benchmark for evaluating FTLE computations. In *2012 IEEE Pacific Visualization Symposium*, pp. 121–128, 2012. doi: 10.1109/PacificVis.2012.6183582 2
- [20] T. McLoughlin, R. S. Laramee, R. Peikert, F. H. Post, and M. Chen. Over Two Decades of Integration-Based, Geometric Flow Visualization. *Computer Graphics Forum*, 29(6):1807–1829, 2010. doi: 10.1111/j.1467-8659.2010.01650.x 1
- [21] R. W. Ogden. *Non-Linear Elastic Deformations*. Dover Publications, Inc., 1997. 2
- [22] R. Peikert and M. Roth. The "Parallel Vectors" operator—a vector field visualization primitive. In *Proceedings of IEEE Visualization 1999*, pp. 263–272, 1999. doi: 10.1109/visual.1999.809896 2
- [23] P. Petersen. *Riemannian Geometry*. Springer-Verlag, 3rd ed., 2016. doi: 10.1007/978-3-319-26654-1 3
- [24] P. Rautek, M. Mlejnek, J. Beyer, J. Troidl, H. Pfister, T. Theußl, and M. Hadwiger. Objective observer-relative flow visualization in curved spaces for unsteady 2D geophysical flows. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):283–293, 2021. doi: 10.1109/tvcg.2020.3030454 2, 3, 4
- [25] P. Rautek, X. Zhang, B. Woschizka, T. Theußl, and M. Hadwiger. Vortex lens: Interactive vortex core line extraction using observed line integral convolution. *IEEE Transactions on Visualization and Computer Graphics*, 30(1):1–11, 2024. doi: 10.1109/TVCG.2023.3326915 2, 3, 7
- [26] J. Sahner, T. Weinkauf, and H.-C. Hege. Galilean invariant extraction and iconic representation of vortex core lines. In *Proceedings of Eurographics/IEEE VGTC Symposium on Visualization 2005*, pp. 151–160, 2005. doi: 10.2312/VisSym/EuroVis05/151-160 2
- [27] S. C. Shadden, F. Lekien, and J. E. Marsden. Definition and properties of Lagrangian coherent structures from finite-time Lyapunov exponents in two-dimensional aperiodic flows. *Physica D: Nonlinear Phenomena*, 212(3):271–304, Dec. 2005. doi: 10.1016/j.physd.2005.10.007 2
- [28] J. Solomon, M. Ben-Chen, A. Butscher, and L. Guibas. As-killing-as-possible vector fields for planar deformation. *Computer Graphics Forum*, 30(5):1543–1552, 2011. doi: 10.1111/j.1467-8659.2011.02028.x 3
- [29] J. Stillwell. *Naive Lie Theory*. Springer Verlag, 2008. doi: 10.1007/978-0-387-78214-0 3
- [30] D. Sujudi and R. Haimes. Identification of swirling flow in 3-d vector fields. In *Proceedings of the 12th Computational Fluid Dynamics Conference*, pp. 792–799, 1995. doi: 10.2514/6.1995-1715 2
- [31] H. Theisel, M. Hadwiger, P. Rautek, T. Theußl, and T. Günther. Vortex criteria can be objectivized by unsteadiness minimization. *Physics of Fluids*, 33:107115, 2021. doi: 10.1063/5.0063817 2
- [32] H. Theisel, T. Weinkauf, H.-C. Hege, and H.-P. Seidel. Feature flow fields. In *Proceedings of the Symposium on Data Visualization (VisSym)*, pp. 141–148, 2005. doi: 10.2312/VisSym/VisSym03/141-148 2
- [33] C. Truesdell and W. Noll. *The Nonlinear Field Theories of Mechanics*. Springer-Verlag, 1965. doi: 10.1007/978-3-662-10388-3_1 2
- [34] J. Weiss. The dynamics of enstrophy transfer in two-dimensional hydrodynamics. *Physica D: Nonlinear Phenomena*, 48(2):273–294, Mar. 1991. doi: 10.1016/0167-2789(91)90088-q 2
- [35] A. Wiebel, C. Garth, and G. Scheuermann. Computation of localized flow for steady and unsteady vector fields and its applications. *IEEE Transactions on Visualization and Computer Graphics*, 13(4):641–651, 2007. doi: 10.1109/TVCG.2007.4293009 2
- [36] X. Zhang, M. Hadwiger, T. Theußl, and P. Rautek. Interactive exploration of physically-observable objective vortices in unsteady 2D flow. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):281–290, 2022. doi: 10.1109/tvcg.2021.3115565 2, 3, 4
- [37] X. Zhang, M. Hadwiger, T. Theußl, and P. Rautek. Enhancing material boundary visualizations in 2d unsteady flow through local reference frame transformations. *Computer Graphics Forum*, 2025. doi: 10.1111/cgf.70128 3, 7
- [38] X. Zhang, P. Rautek, and M. Hadwiger. Vortextransformer: End-to-end objective vortex detection in 2d unsteady flow using transformers. *Computer Graphics Forum*, p. e70042, 2025. doi: 10.1111/cgf.70042 2