

Vortex Lens: Interactive Vortex Core Line Extraction using Observed Line Integral Convolution

Peter Rautek, Xingdi Zhang, Bernhard Woschizka, Thomas Theußl, Markus Hadwiger

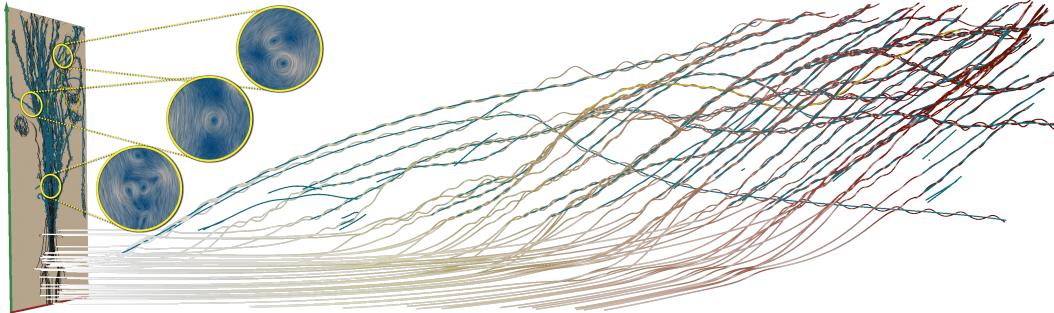


Fig. 1: **Vortex structures** in an unsteady 2D flow field identified and extracted via interactive exploration. The plane (left) shows vortex core lines (green) and path lines (black). Path lines are seeded close to critical points. The horizontal axis is time: Core lines and path lines (white-red) in space-time. The vortex lens regions (yellow insets) show vortex structures visualized with *Observed Line Integral Convolution* (LIC), i.e., LIC computed from an observed velocity field relative to a locally optimal reference frame in each vortex lens.

Abstract—This paper describes a novel method for detecting and visualizing vortex structures in unsteady 2D fluid flows. The method is based on an interactive local reference frame estimation that minimizes the observed time derivative of the input flow field $\mathbf{v}(x,t)$. A locally optimal reference frame $\mathbf{w}(x,t)$ assists the user in the identification of physically observable vortex structures in *Observed Line Integral Convolution* (LIC) visualizations. The observed LIC visualizations are interactively computed and displayed in a user-steered vortex lens region, embedded in the context of a conventional LIC visualization outside the lens. The locally optimal reference frame is then used to detect observed critical points, where $\mathbf{v} = \mathbf{w}$, which are used to seed vortex core lines. Each vortex core line is computed as a solution of the ordinary differential equation (ODE) $\dot{\mathbf{v}}(t) = \mathbf{w}(w(t), t)$, with an observed critical point as initial condition $(w(t_0), t_0)$. During integration, we enforce a strict error bound on the difference between the extracted core line and the integration of a path line of the input vector field, i.e., a solution to the ODE $\dot{\mathbf{v}}(t) = \mathbf{v}(v(t), t)$. We experimentally verify that this error depends on the step size of the core line integration. This ensures that our method extracts Lagrangian vortex core lines that are the simultaneous solution of both ODEs with a numerical error that is controllable by the integration step size. We show the usability of our method in the context of an interactive system using a lens metaphor, and evaluate the results in comparison to state-of-the-art vortex core line extraction methods.

Index Terms—Flow visualization, vortex detection, objectivity, observers, reference frames, Lie algebras, visual lens metaphors

1 INTRODUCTION

The detection, visualization, and analysis of vortices in unsteady fluid flows is an important well-established problem in flow visualization and continuum mechanics [14, 17, 28]. However, one crucial problem in this context is that velocity vector fields always depend on the chosen reference frame, or *observer*, relative to which they were measured or computed. For this reason, it is important that vortex detection methods are independent of suitable classes of reference frame motion [14]. Our work belongs to the class of *objective* approaches [17], which are invariant with respect to any rigid motion of the reference frame [46].

Recently, several methods have been proposed that compute “optimal” reference frames through optimization, with the goal of making the observed flow field “as steady as possible.” The latter is quantified

by computing a time derivative with respect to the optimal reference frame [16]. Different variants either compute optimal frames in small spatial neighborhoods [2, 12], or in an entire domain of interest [16, 35]. However, all of these methods have one drawback in common: By minimizing over a whole region or neighborhood (either locally or globally), the “steadiness” of the resulting observed flow field is often not specifically optimal for any potential (but unknown) vortex core line, i.e., for one specific point at each time (in 2D flow). In this sense, the above approaches are only optimal when the time derivative measure can be minimized to exactly zero. However, in general, and in particular for real-world data, although the time derivative does become small, it remains non-zero. This implies that often no potential vortex core line location lies exactly on a point where the observed vector field is really a steady vector field. Furthermore, all previous methods compute the optimal reference frames offline and not interactively. This makes optimization parameters costly and hard to steer by the user.

In this paper, we propose the first approach for interactive reference frame optimization in unsteady flow fields that is interactively steered by the user, without any pre-computation. User interaction is built around a *vortex lens* metaphor, denoting a current region of interest, in both space and time, in which a locally optimal reference frame is computed interactively. However, in contrast to earlier work we do not compute an optimal reference frame first, and then only afterward compute vortex core lines. Instead, we interactively perform a *joint* optimization: We jointly optimize for (1) an unknown reference frame together with (2) an unknown vortex core line candidate, such

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

• Thomas Theußl is with King Abdullah University of Science and Technology (KAUST), Core Labs, Thuwal, 23955-6900, Saudi Arabia. E-mail: thomas.theussl@kaust.edu.sa.

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

that both together are optimal in a certain sense. We quantify optimality by an objective function that combines both (1) a *critical point criterion* (a vortex core line should always go through a critical point, i.e., a zero of the observed velocity field), and (2) an *observed time derivative criterion* (the observed field should be as steady as possible).

We show that in this way, (1) our method finds better optimal vortex core line candidates than previous methods; and that (2) the vortex lens user interaction enables steering the optimization for optimal results in a natural way. While the user interactively changes the space-time position and size of the vortex lens, we asynchronously jointly optimize both the reference frame and a potential vortex core line candidate, using an iterative approach based on the often-used proximal Alternating Direction Method of Multipliers (ADMM) algorithm [31]. Moreover, we build on previous work that describes the time-dependent motion of reference frames using the notion of observer velocity fields [16, 35, 53].

1.1 Mathematical Framework

Following the mathematical framework described by Zhang et al. [53], we describe a Euclidean reference frame by a time-dependent observer velocity field $\mathbf{w}(x, t)$. The field \mathbf{w} is in fact very simple, because it describes an infinitesimal isometry, i.e., it is a Killing vector field [16] whose integration gives the corresponding isometry. In \mathbb{R}^2 , isometries have three degrees of freedom per time step (two for translation, one for rotation), and all infinitesimal isometries form a 3D vector space of vector fields in \mathbb{R}^2 [53]. Thus, any observer field \mathbf{w} can be expressed as

$$\mathbf{w}(x, t) = a(t) \mathbf{e}_1(x) + b(t) \mathbf{e}_2(x) + \omega(t) \mathbf{e}_3(x), \quad (1)$$

with time-dependent coefficient functions $t \mapsto (a, b, \omega)(t)$; $a, b, \omega \in \mathbb{R}$. Although \mathbf{w} is an unsteady vector field, we choose the three basis vector fields \mathbf{e}_i as linearly-independent *steady* vector fields in \mathbb{R}^2 . See Sec. 4.2.

1.2 Joint Observer and Vortex Core Line Optimization

Our goal is to avoid any pre-computation of reference frames, in contrast to all similar prior work on optimal reference frames [12, 16, 35, 53], and instead interactively minimize the functional in Eq. 2 below for each time step t , jointly comprising a *critical point criterion* and an *observed time derivative criterion* over a spatial vortex lens region $R(t) \subset \mathbb{R}^2$.

For each time step t , we want (1) a point $c(t)$ that lies on a *vortex core line candidate*, ultimately given by a curve $t \mapsto c(t) \in \mathbb{R}^2$; and (2) an observer velocity field $\mathbf{w}(x, t)$, ultimately known for all time steps t . For a given vortex lens region $R(t)$, we *jointly* compute minimizers of

$$\min_{c(t), \mathbf{w}(x, t)} \|(\mathbf{v} - \mathbf{w})(c(t), t)\| + \frac{\mu}{A} \int_{R(t)} \|L_{\mathbf{w}}(\mathbf{v} - \mathbf{w})(x, t)\| dA. \quad (2)$$

Both the position $c(t)$ and the observer field $\mathbf{w}(x, t)$ are unknown variables that we optimize for each time t . The region $R(t)$ with area A is centered around $c(t)$; $\mu \in \mathbb{R}$ is a relative weight. Both terms in the objective function above use the L_1 norm, which we implement by using proximal ADMM [31] to obtain the minimizer of Eq. 2. To make the optimization interactive, we optimize time steps progressively, iterating from one time to the next. The curve $t \mapsto c(t)$ and the optimal observer $t \mapsto \mathbf{w}(x, t)$ are thus only known in a growing time window $[t_0, t]$, with t progressively increasing from some user-selected start time t_0 .

Eq. 2 includes a *critical point criterion* (vanishing observed velocity)

$$(\mathbf{v} - \mathbf{w}) = 0. \quad (3)$$

This means that a *vortex core line goes through observed critical points* in the observed velocity field, relative to the reference frame \mathbf{w} of the lens. The observed velocity at these points vanishes, as in steady flows. The second term in Eq. 2 denotes an *observed time derivative criterion*

$$L_{\mathbf{w}}(\mathbf{v} - \mathbf{w}) = 0. \quad (4)$$

This means that *along a vortex core line the observed time derivative vanishes*, relative to the optimal reference frame \mathbf{w} (see also Eq. 10).

The overall goal of Eq. 2 is therefore that the curve $t \mapsto c(t)$ passing through observed critical points will map to a single stationary point c

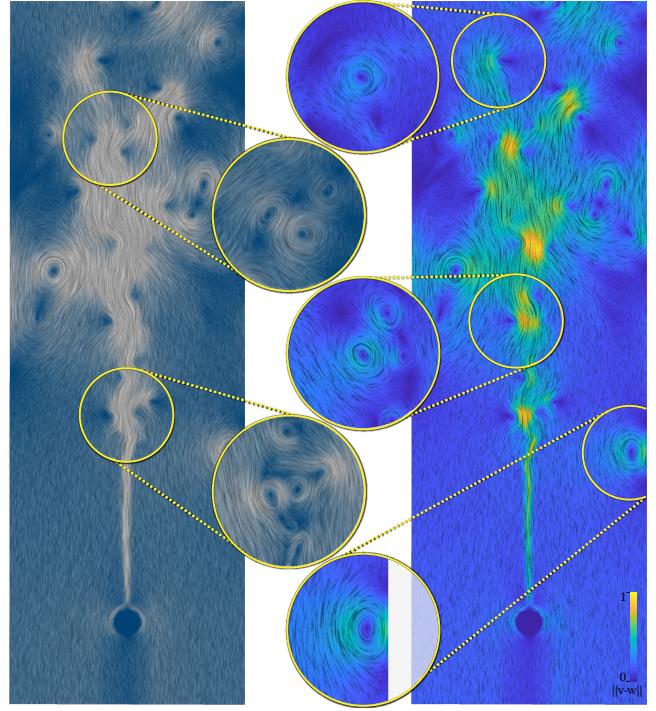


Fig. 2: Observed Line Integral Convolution (LIC). The insets depict observed LIC images in five vortex lens regions. They are computed from *observed velocities*, relative to a different locally-optimal observer for each lens. Although for unsteady flows LIC only shows instantaneous snapshots that often do not depict vortex structures correctly, *observed LIC* with a suitably optimized observer solves this problem locally, by visualizing observed velocity fields that are *as steady as possible*. However, critical points in observed velocity fields, which become clearly visible in observed LIC images, reveal vortex structures. Color-coding on the right: observed velocity magnitude (in insets) vs. original velocity magnitudes.

in the optimal reference frame, as in steady flows. We note that this has some similarities to feature flow fields [44], but we enforce physical consistency by optimizing for a field \mathbf{w} that is guaranteed to describe a physical reference frame motion [53], because \mathbf{w} is a Killing field.

Given $c(t)$ and $\mathbf{w}(x, t)$ as minimizers of the objective function in Eq. 2, we call the curve $t \mapsto c(t)$ an *optimal vortex core line candidate*. It is only a candidate, because there is no term in the functional above that optimizes for actual swirling motion around the core line candidate. However, this is then simply confirmed afterward by computing different vortex criteria on every candidate line, together with interactive visualization of these criteria, e.g., color-coding complex eigenvalues of the observed velocity gradient $(\nabla \mathbf{v} - \nabla \mathbf{w})$ [53] for each candidate $c(t)$.

1.3 Observed Line Integral Convolution

A crucial ingredient of our framework is the user's interaction with and visual inspection of *observed velocity fields* in an interactively steered vortex lens region $R(t)$, using *observed line integral convolution (LIC)* visualizations. Given the observer field $\mathbf{w}(x, t)$ in a vortex lens for time t , within the respective region $R(t)$ we compute standard line integral convolution of the corresponding observed velocity field

$$(\mathbf{v} - \mathbf{w}). \quad (5)$$

Despite the simplicity of this equation, because the field \mathbf{w} has been optimized to minimize the observed time derivative as described above, the corresponding observed velocity field is as steady as possible, and its observed critical points reveal structures such as vortices as in steady flows. Fig. 2 illustrates this for five example vortex lens regions.

Moreover, because the computation of the observer field \mathbf{w} is *objective*, as in prior work [12, 16, 53], in any given vortex lens region $R(t)$ our observed LIC is an *objective visualization*: It is independent of the reference frame for which the input flow field \mathbf{v} was measured or simulated, and observed LIC visualizations will always look the same.

1.4 Vortex Lens in Space-Time

While above we have introduced the vortex lens concept simply as a time-dependent spatial region $R(t)$, an important idea of our framework is that the vortex lens is automatically, and progressively, extended into space-time, from an original lens region $R(t_0)$ from some start time t_0 .

The progressive computation of the observer field $\mathbf{w}(x, t)$ allows us to *advection* the current lens region $R(t)$ forward in time to obtain the corresponding lens region $R(t + \tau)$ in the next time step $t + \tau$. We will denote the flow map of the vector field \mathbf{w} by $\phi_a^{t_b}$, which corresponds to mapping any set of spatial positions in the flow domain from time t_a to the corresponding position under advection by \mathbf{w} at time t_b , and define

$$R(t + \tau) = \phi_t^{t+\tau}(R(t)). \quad (6)$$

It is this “self-advection” of the vortex lens under the flow map of the observer field optimized for the lens region that allows us to track vortex core lines, and the corresponding observed LIC, through time.

We note that since the field \mathbf{w} is always a Killing field, its flow map ϕ comprises isometries, and thus the self-advection of the vortex lens is a rigid motion and no deformations are introduced. Furthermore, since we use circular vortex lenses, the rigid motion reduces to a translation.

2 RELATED WORK

Vortex detection. The detection and visualization of vortices in fluid flows is a very important topic in flow visualization and fluid mechanics. Computational methods can be categorized as either region-based or line-based [14]. Well-known examples for region-based methods include the criteria of Okubo [30], Weiss [50], Hunt [21], and Jeong and Hussain [22]. Examples for line-based methods are Sujudi and Haimes [41], the computation of feature flow fields for critical points [44], the parallel vectors operator of Peikert and Roth [32], and tracking of swirling streamlines [4, 43, 45]. The LAVD method of Haller et al. [19] detects vortices both as regions as well as a vortex core line located at the isolated maximum of the LAVD scalar field. Other vortex detection methods include those based on vector field decomposition [6], the geometric method by Xie et al. [52], the integration based methods by Globus et al. [11], Wiebel et al. [51], and Sadlo et al. [37], the predictor-corrector method by Banks and Singer [3], and vortex boundary detection using machine learning methods [5].

Vortex visualization. Although many techniques for visualizing fluid flow developed over the years, like Line Integral Convolution (LIC) [9] and its improvements [27, 39, 40], dense image based methods [26, 47, 48], texture based methods [23], or streamline-based visualization [24], do not deal directly with the detection of vortices. These structures play a crucial role in the interpretation of the resulting visualization, as they define the topological structure of the flow [33, 34]. A summary of the state of the art in vortex extraction specifically can be found in Günther and Theisel [14]. A recent overview dealing with topology-based flow visualization in detail is given by Bujack et al. [8].

Reference frame invariance. It has been recognized that the detection of features such as vortices should be independent of any particular reference frame relative to which a flow field is given [14]. Early approaches typically focused on Galilean invariance [38, 49], including more recent advances [7]. Going beyond Galilean motion, independence of any rigid reference frame motion, including rotations, is known as *objectivity* or frame indifference, which is a traditional topic in continuum mechanics [1, 10, 20, 29, 46] that has received significant recent attention since the seminal work of Haller in 2005 [17], and is still a very active topic of research [12, 16, 18, 19, 42]. Other kinds of invariance, such as rotation invariance [13] have also been defined.

Reference frame optimization. Recently, several methods have been proposed that find good reference frames through (non-interactive) optimization. Günther et al. [12], Baeza Rojo and Günther [2], and Theisel et al. [42] propose local optimization schemes that only optimize over individual small neighborhoods. Hadwiger et al. [16] and Rautek et al. [35] propose global optimization schemes over an entire domain of interest. Zhang et al. [53] build on these optimization results for the interactive exploration of flow fields. Kim and Günther use machine learning methods for robust reference frame extraction [25].

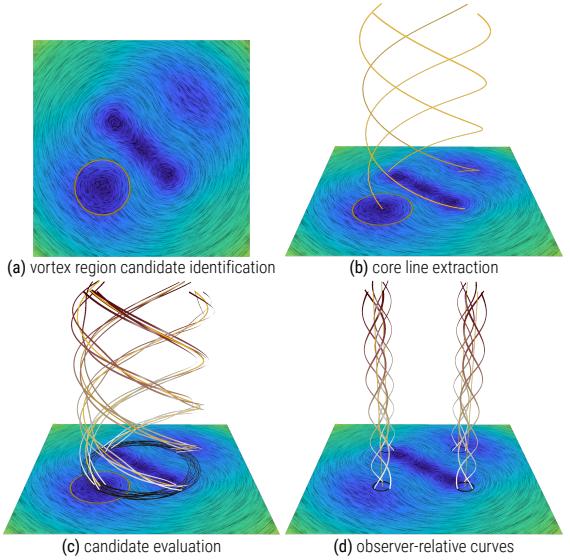


Fig. 3: **Interactive exploration.** (a) The user starts by selecting a vortex lens region in the flow, determining space-time position and region size (radius of the lens). We interactively perform a reference frame optimization for the lens region. (b) We compute a vortex core line starting from the lens region. For interactivity, the core line extraction is running asynchronously in the background. (c) The user can immediately evaluate the quality of vortex core line candidates by interactively seeding integral curves and color-coding a variety of vector field properties. (d) The vector field describing the optimal reference frame motion associated with one core line can immediately be used for observer-relative visualizations.

In this paper, we focus on a line-based approach for vortex detection, where we optimize for the best vortex core line candidate in the region corresponding to a vortex lens. We use a reference frame optimization approach and compare to other methods using reference frame transformations. As pointed out by Bujack et al. [8], most methods for vortex detection in unsteady flows use reference frames, either explicitly or implicitly, to compensate for ambient motion. We compare our approach to both kinds of methods. We categorize our approach according to Bujack et al. [8] as: *coinciding with the steady case, segmenting areas of coherent behavior, Galilean invariant, and objective*. Although our approach is not strictly *Lagrangian* per their categorization, unlike some other approaches analyzed by them [8], we explicitly control the error of our solution as it deviates from a true Lagrangian solution.

3 INTERACTIVE EXPLORATION USING THE VORTEX LENS

Our approach relies on a user in the loop to identify candidates for vortex regions by interactively steering the *vortex lens*. The lens comprises a disk-shaped region $R(t)$, of a user-controlled radius r , with a different region per time t . However, the user always only selects or changes the vortex lens position and size for any desired time step t_0 , and our framework automatically, and asynchronously, propagates the region $R(t_0)$ throughout time using Eq. 6. In this way, $R(t_0)$ is progressively grown in time to obtain a *space-time* vortex lens $R([t_0, t])$, with t increasing automatically as the optimization of the observer field $\mathbf{w}(x, t)$ within the region $R([t_0, t])$ progresses from time step t_i to time step t_{i+1} .

Starting with an initial region $R(t_0)$, as in Fig. 3(a), the automatic extension of $R(t_0)$ to $R([t_0, t])$ in space-time is what enables the automatic extraction of vortex core lines *within* the space-time domain $R([t_0, t])$ of the vortex lens, as shown in Fig. 3(b). Because the observer field \mathbf{w} is specifically computed just for the vortex lens region, self-adverting the lens via Eq. 6 is necessary in order to be able to follow the observed velocity field ($\mathbf{v} - \mathbf{w}$) through time. The same holds for the user manually seeding path lines in the vortex lens region, as depicted in Fig. 3(c). Finally, we can also perform a fully interactive *reference transformation* within the space-time region $R(t)$ of any given vortex lens, which results in “straightening out” vortex core lines and path lines swirling around them within the lens, as depicted in Fig. 3(d).

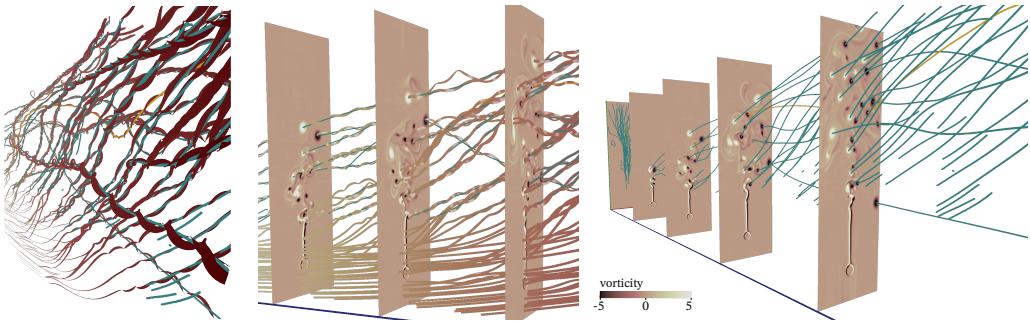


Fig. 4: **Vortex core lines and path lines in space-time.** Left: path lines closely swirling around vortex cores. Middle: Path lines and core lines only. Planes depict time slices; time is horizontal. Right: vortex cores only.

3.1 Basic Fields for Observer-Relative Visualization

In order to (1) aid the user in finding an initial vortex lens region $R(t_0)$, and (2) help with finding vortex cores and analyzing the observed flow in the vortex lens, we provide a variety of visualizations of the properties of different vector fields. We visualize scalar properties such as the λ_2 criterion [22], and provide dense vector field visualizations such as LIC as well as computing path lines from interactively specified seed points. In contrast to typical flow visualizations, we provide these capabilities for several different vector fields that are crucial components of interactive vortex identification and verification:

- The input flow field \mathbf{v} . Naturally, all visualizations that we provide are meaningful for the input field, with the usual reservation that LIC images are only instantaneous snapshots in time and do not correspond to actual particle motion for unsteady flows \mathbf{v} .
- The observed relative velocity field $(\mathbf{v} - \mathbf{w})$ in the vortex lens region $R(t)$, embedded in the original input flow \mathbf{v} for context. Visualizing instantaneous *observed* LIC images and color-coding properties of individual time steps of this field is meaningful, even for *unsteady* flows. However, computing path lines is not: No particles are advected by the unsteady field $(\mathbf{v} - \mathbf{w})$ [53] if $\mathbf{w} \neq 0$. The particle trajectories depicted in Fig. 3(c) are computed from the input field \mathbf{v} , not from the field $(\mathbf{v} - \mathbf{w})$, although they were seeded in a vortex lens guided by the observed LIC of $(\mathbf{v} - \mathbf{w})$.
- The input field \mathbf{v} fully transformed to the reference frame given by the observer field \mathbf{w} in the vortex lens region $R(t)$. We denote this transformed velocity field by $\phi_t^*(\mathbf{v} - \mathbf{w})$. The map ϕ_t^* is the pullback of the isometry ϕ_t [53], corresponding to the integrated field \mathbf{w} . Because it is an isometry, ϕ_t is a translation and rotation, and ϕ_t^* applies a rotation to the vectors $(\mathbf{v} - \mathbf{w})$. In this transformed field, path lines are again meaningful: Physical (massless) particles move along the corresponding integral curves in the reference frame given by \mathbf{w} . The particle trajectories in Fig. 3(d) are computed from the transformed velocity field $\phi_t^*(\mathbf{v} - \mathbf{w})$.

While the distinction between the latter two points above might seem to be minor, in terms of physical meaning of vector fields it is in fact crucial. In our interactive vortex exploration, they imply that the user needs to be aware of these reference frame-dependent subtleties: We allow for simultaneous visualization and comparison of path lines of the field \mathbf{v} and the field $\phi_t^*(\mathbf{v} - \mathbf{w})$, respectively, but not of the field $(\mathbf{v} - \mathbf{w})$, although the optimization problem of Eq. 2 uses the latter vector field instantaneously, and observed LIC images in a vortex lens show the field $(\mathbf{v} - \mathbf{w})$. Nevertheless, visualizing instantaneous properties of the field $(\mathbf{v} - \mathbf{w})$, in particular observed critical points, is meaningful and important, and significantly helps steer the vortex lens, as follows.

3.2 Observed Velocity Fields and Observed LIC

Denoting the input flow field by \mathbf{v} and the observer field by \mathbf{w} , we visualize the observed velocity magnitude by color-coding $\|\mathbf{v} - \mathbf{w}\|$ on individual time slices in order to help identify critical points in the field $(\mathbf{v} - \mathbf{w})$. This simple visual approach is well-suited to help identify vortex region candidates. In the left insets depicted in Fig. 2, we show observed LIC images for a local optimal observer per vortex lens inset.

In the right insets in Fig. 2, observed velocity magnitude is color-coded. All observed velocity fields visualized in Fig. 2 are from the same input field \mathbf{v} , but relative to different observer fields \mathbf{w} . We want to highlight that the observed velocity depends on the \mathbf{w} that is the outcome of a local optimization. When the user selects a point, our system computes the observer that is locally optimal for the lens region around this point. This approach of embedding local observers within the context of the original global observer is very suitable for interactive exploration. The field $(\mathbf{v} - \mathbf{w})$ allows identifying meaningful observed critical points without having to perform a global reference frame transformation first.

3.3 Vortex Core Line Candidates

Our interactive optimization gives a *vortex core line candidate* $t \mapsto c(t)$ by minimizing the objective function given in Eq. 2. However, although the goal is to find vortex core lines, we do not explicitly optimize for any specific vortex criterion related to swirling particle behavior. Therefore, each vortex core line candidate must be verified afterward. To help in this task, any vortex criterion resulting in a scalar field, such as λ_2 [22], can be color-coded interactively in spatial regions, for any time step t , as well as on the current vortex core line candidate curve. In order to obtain a vortex core line candidate, we follow these major steps:

- We obtain each vortex core line candidate $t \mapsto c(t)$, together with a reference frame $t \mapsto \mathbf{w}(x, t)$, from the iterative joint optimization described in Sec 5. We enforce that the candidate described by the curve $t \mapsto c(t)$ obtained via optimization is as close as possible to a world line (an integral curve) $t \mapsto w(t)$ of the observer field \mathbf{w} (see also the discussion in Sec. 4.1). For any time step t_i with $i > 0$, we initialize the optimization of the unknown point $c(t_i)$ with the position obtained via an integral curve of the observer field \mathbf{w} that is seeded at the previous point $c(t_{i-1})$, and integrated forward in time to obtain a point $w(t_i)$. We then set $c(t_i) = w(t_i)$ and start optimizing the position $c(t_i)$ jointly with the field $\mathbf{w}(t_i)$. That is, both the point $c(t_i)$ and the field $\mathbf{w}(t_i)$ change until we have that $c(t_i)$ is an *observed critical point* relative to the field \mathbf{w} .
- A goal for any vortex core line is for it to be Lagrangian, i.e., that it is a path line $t \mapsto v(t)$ of the input flow \mathbf{v} . In light of the above, this means that we are searching for an unknown observer \mathbf{w} such that a specific observer world line $t \mapsto w(t)$ is identical to a path line $t \mapsto v(t)$. This can only be true if the critical point criterion in Eq. 2 is zero on the world line $t \mapsto w(t)$. We therefore analyze the error given by the deviation of the three curves $c(t)$, $w(t)$, and $v(t)$. We can enforce convergence toward the same trajectory for all three curves by decreasing the integration step size (Sec. 6.1).
- Finally, the desired behavior of particles swirling around the vortex core line candidate must be verified, using any one of many available criteria (see below), in order to determine whether the candidate in fact qualifies as a vortex core line.

Again, the second point above is subtle: The goal is to search for a vortex core line candidate that is an actual particle trajectory, around which we will subsequently verify that other particles exhibit swirling behavior. However, without the optimization finding the minimizer of Eq. 2 we would not have found this core line candidate, except by exhaustive search and extensive verification.

We cannot simply search for critical points in the input field \mathbf{v} , because they are only meaningful with respect to a reference frame that allows transforming the input field into a steady observed field. For this, however, we have to compute the reference frame \mathbf{w} first.

3.4 Observed Critical Point Analysis

To determine swirling behavior around a found vortex core line candidate, we color-code the corresponding integral curve by whether the eigenvalues of the observed velocity gradient ($\nabla\mathbf{v} - \nabla\mathbf{w}$) are complex.

To do this, we check for a negative sign of the discriminant $\Delta = b^2 - 4c$ of the characteristic polynomial $\lambda^2 - b\lambda + c$ of the observed velocity gradient. (We note that the coefficients b, c are the standard notation for the coefficients of a quadratic polynomial, but have nothing to do with the (a, b, ω) values that we use to describe observer fields \mathbf{w} .)

Denoting the components of the tensors $\nabla\mathbf{v}$ and $\nabla\mathbf{w}$ with row i , column j by $\nabla_j v^i$ and $\nabla_j w^i$, respectively, we have $b = \nabla_1 v^1 + \nabla_2 v^2$, $c = \nabla_1 v^1 \nabla_2 v^2 - (\nabla_2 v^1 - \nabla_2 w^1)(\nabla_1 v^2 - \nabla_1 w^2)$, for the observed velocity gradient. The observed discriminant in expanded form is therefore

$$\begin{aligned}\Delta &= (\nabla_1 v^1 + \nabla_2 v^2)^2 - 4(\nabla_1 v^1 \nabla_2 v^2 - (\nabla_2 v^1 - \nabla_2 w^1)(\nabla_1 v^2 - \nabla_1 w^2)), \\ &= (\nabla_1 v^1 + \nabla_2 v^2)^2 - 4(\nabla_1 v^1 \nabla_2 v^2 - \nabla_2 v^1 \nabla_1 v^2 + \\ &\quad \nabla_2 v^1 \nabla_1 w^2 + \nabla_1 v^2 \nabla_2 w^1 - \nabla_2 w^1 \nabla_1 w^2).\end{aligned}\quad (7)$$

We note that the first, “trace-squared,” term does not depend on the observer field \mathbf{w} , but the second, “four times determinant,” term does.

4 OBSERVER FIELDS

We build on previous work that describes the time-dependent motion of reference frames using the notion of observer velocity fields [16, 35, 53]. In the following, we give the individual definitions that we rely on heavily in our interactive framework and in the rest of the paper, in particular for the optimization process described in Sec. 5.

An observer field is simply a velocity field that describes the (relative) motion of a reference frame by an entire unsteady vector field. In this work, we use an observer field denoted by \mathbf{w} , which describes an *infinitesimal isometry*, i.e., the derivative of a rigid motion. Such a vector field is also called a Killing vector field [16]. This setup unifies and simplifies many observer-relative computations, and enables fully interactive transformations into a new reference frame, even while the reference frame is changing [53]. In our framework, we rely on this capability to interactively visualize the input field relative to the reference frame that is currently being optimized for the vortex lens region.

An observer field \mathbf{w} enables especially simple computations with reference frames, because the field \mathbf{w} is fully determined by a time-dependent function $t \mapsto (a, b, \omega)(t)$, where $a, b, \omega \in \mathbb{R}$, determining the field \mathbf{w} via Eq. 1. However, once the field \mathbf{w} is known, it is advantageous and often simpler to use \mathbf{w} *like any other velocity field* in equations, without always having to specify the semantics of an observer.

4.1 Observer World Lines

A world line $t \mapsto w(t)$ of an observer field \mathbf{w} is a solution of the ODE $\dot{w}(t) = \mathbf{w}(w(t), t)$, with an initial value $w(t_0)$, i.e., some position on the path line $t \mapsto w(t)$ at time $t = t_0$. In our framework, observer world lines are important, because our goal during optimization is to jointly search for an observer field \mathbf{w} and a vortex core line candidate $t \mapsto c(t)$, such that the latter is a world line of the computed observer field. That is, ideally we have $t \mapsto c(t) = w(t)$. This will mean that the observed vortex core line is a stationary point in a steady transformed velocity field $\phi_t^*(\mathbf{v} - \mathbf{w})$. However, if the actual minimum obtained in Eq. 2 is not exactly zero, the above will only be approximately true. Our approach therefore tracks and iteratively minimizes the deviation between the two curves $c(t)$ and $w(t)$. The difference between these two curves is a measure for the optimality of the detected vortex core line candidate. In space-time visualizations, the above means that a vortex core line with $t \mapsto c(t) = w(t)$ is perfectly vertical. See Fig. 3(d).

4.2 Observer Basis Vector Fields

The representation of an observer field \mathbf{w} in \mathbb{R}^2 as a time-dependent function $t \mapsto (a, b, \omega)(t)$ and its computation via Eq. 1 rely on a choice of three linearly-independent *basis observer fields* [53] in \mathbb{R}^2 . Apart from the requirement of linear independence, each basis field must be a Killing vector field [16], i.e., the derivative of a rigid motion, but is otherwise arbitrary. However, it is helpful to choose basis fields corresponding to an intuitive physical meaning, as described below.

We choose the following basis observer fields, where the vectors at any point $p = (x, y) \in \mathbb{R}^2$, in Cartesian coordinates, are given by

$$\mathbf{e}_1(x, y) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mathbf{e}_2(x, y) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \mathbf{e}_3(x, y) = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \quad (8)$$

In the definition above, we define spatial coordinates $(x, y) \in \mathbb{R}^2$ relative to the origin at the center of mass of the flow domain, given by the average of all spatial positions within the domain. The three vector fields \mathbf{e}_i are *linearly independent* as vector fields in \mathbb{R}^2 (not as individual vectors at just a single point). Therefore, they comprise a *basis* of the three-dimensional vector space of Killing vector fields in \mathbb{R}^2 , corresponding to the derivatives of all possible rigid motions in \mathbb{R}^2 . The specific basis vector fields given above are chosen such that the three parameters (a, b, ω) have the physical meaning that $(a, b)^T$ are the *translation velocity*, and ω is the *angular velocity*, respectively, of the reference frame motion described by the velocity field \mathbf{w} in Eq. 1.

4.3 Observed Time Derivative

For the computation and use of reference frames for the observation of unsteady input flow fields \mathbf{v} , the most important ingredient is a time derivative of a given velocity field \mathbf{v} relative to a given observer field \mathbf{w} . We call this time derivative the *observed time derivative* [16] relative to an arbitrary reference frame motion given by the vector field \mathbf{w} .

Relative to an arbitrary given reference frame motion \mathbf{w} , the observed time derivative of an arbitrary unsteady input vector field \mathbf{v} is

$$\frac{\partial \mathbf{v}^*}{\partial t}(x^*, t) = \phi_t^*(L_{\mathbf{w}}(\mathbf{v} - \mathbf{w})(x, t)). \quad (9)$$

Here, the map ϕ_t^* denotes the *pullback* of the isometry ϕ corresponding to the observer transformation, as described in Sec. 3.1.

The observed time derivative term $L_{\mathbf{w}}(\mathbf{v} - \mathbf{w})$ is given by [16]

$$L_{\mathbf{w}}(\mathbf{v} - \mathbf{w}) = \frac{\partial \mathbf{v}}{\partial t} - \frac{\partial \mathbf{w}}{\partial t} + \nabla \mathbf{v}(\mathbf{w}) - \nabla \mathbf{w}(\mathbf{v}). \quad (10)$$

In terms of the observer field \mathbf{w} determined by a time-dependent function $t \mapsto (a, b, \omega)(t)$ and Eq. 1, the individual terms above are

$$\begin{aligned}\mathbf{w}(x, t) &= a(t) \mathbf{e}_1(x) + b(t) \mathbf{e}_2(x) + \omega(t) \mathbf{e}_3(x), \\ \nabla \mathbf{w}(x, t) &= a(t) \nabla \mathbf{e}_1(x) + b(t) \nabla \mathbf{e}_2(x) + \omega(t) \nabla \mathbf{e}_3(x), \\ \frac{\partial \mathbf{w}}{\partial t}(x, t) &= \frac{da(t)}{dt} \mathbf{e}_1(x) + \frac{db(t)}{dt} \mathbf{e}_2(x) + \frac{d\omega(t)}{dt} \mathbf{e}_3(x).\end{aligned}\quad (11)$$

The velocity gradients $\nabla \mathbf{e}_i$, of the basis observer fields given in Eq. 8, required by the expansion of $\nabla \mathbf{w}$ above, are

$$\nabla \mathbf{e}_1 = 0, \quad \nabla \mathbf{e}_2 = 0, \quad \nabla \mathbf{e}_3 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}. \quad (12)$$

In this way, by the choice of basis in Eq. 8, the parameter $\omega(t)$ directly attains the physical meaning of *instantaneous angular velocity* at time t .

5 ITERATIVE JOINT OPTIMIZATION

We jointly compute optimal vortex core line candidates and reference frame motions asynchronously during user interaction. Fig. 5 illustrates our iterative optimization algorithm. As shown in Fig. 5 ①, we start from a region $R(t_i)$ that is identified by the user to be a good candidate for a vortex region at some time t_i , given via a start point $c_0(t_i)$ in space

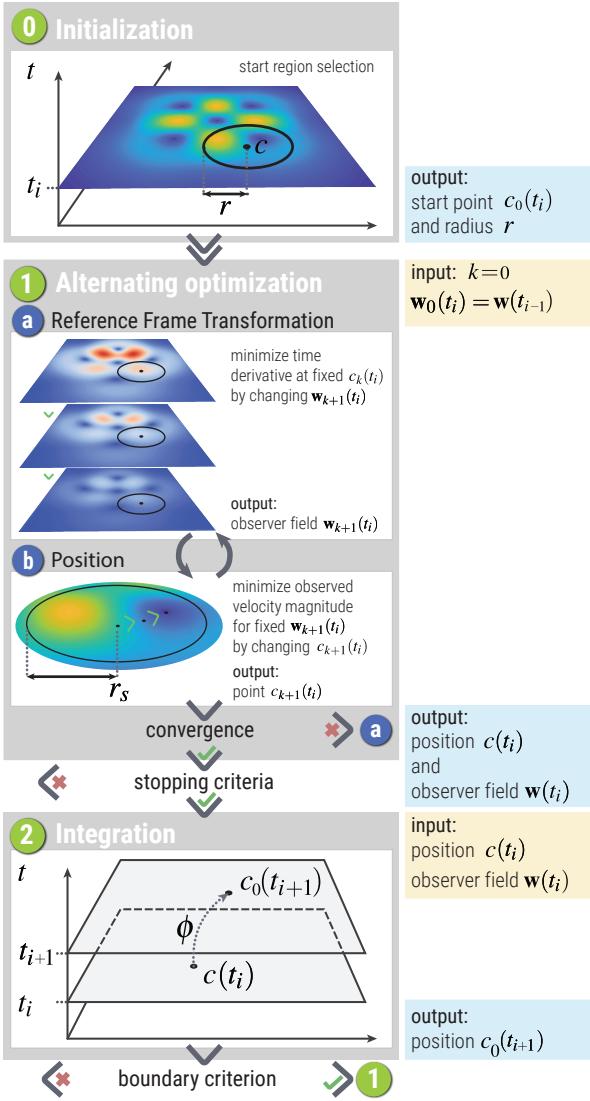


Fig. 5: Iterative joint optimization. To jointly optimize for a vortex core line candidate $c(t)$ together with a reference frame motion described by an observer field $w(t)$, we asynchronously perform ADMM iterations during user interaction, minimizing Eq. 2 in the vortex lens region $R(t)$.

and time, and a user-specified vortex lens radius r . See also Fig. 3(a). Then, vortex core line candidates (Fig. 3(b)) are computed iteratively.

We compute the time-dependent $t \mapsto c(t)$ and $t \mapsto w(t)$ by iterating from time t_i to time t_{i+1} . Each time t_i is computed in two major steps:

1. The joint optimization of a reference frame transformation $w(t_i)$ and a vortex core line position $c(t_i)$ using the ADMM algorithm in two alternating phases (Sec. 5.1). See Fig. 5 ①.
2. The integration of the field w , starting from the optimal point $c(t_i)$, at time t_i , to obtain the start point $c_0(t_{i+1})$ for the next time t_{i+1} (Sec. 5.3). See Fig. 5 ②. If none of the stopping criteria (Sec. 5.2) are met, the algorithm continues with step (1) for time $t = t_{i+1}$.

5.1 Alternating Optimization

For each time t_i , we need to jointly optimize for a reference frame $w(t_i)$ and a point $c(t_i)$ to minimize Eq. 2. In step (1), depicted in Fig. 5 ①, we iteratively alternate between two ADMM optimization phases until they converge to a local optimum, enumerating the iterations of these phases with an index k . In phase (1a), depicted in Fig. 5 ① a, we hold the position $c_k(t_i)$ constant, and optimize for the reference frame $w_{k+1}(t_i)$ that minimizes the time derivative in Eq. 2. In phase (1b), depicted in Fig. 5 ① b, we hold the reference frame $w_{k+1}(t_i)$ constant, and minimize the observed velocity magnitude in Eq. 2 by moving the point

$c_{k+1}(t_i)$ around the previous point $c_k(t_i)$ within a search radius r_s .

The two phases (1a) and (1b) are alternated and coupled by the proximal ADMM (alternating direction method of multipliers) algorithm [31] until they converge, or a maximum iteration count is reached.

Proximal ADMM. The proximal ADMM algorithm solves a convex, possibly non-smooth, optimization problem [31, p. 153]. The general problem can be stated, for some functions f and g , as

$$\operatorname{argmin}_{\mathbf{x}} f(\mathbf{x}) + g(\mathbf{x}). \quad (13)$$

In our optimization algorithm, the function f corresponds to the integral of the observed time derivative (see Eq. 2 and Eq. 10) over the vortex lens region $R(t)$, of radius r centered around the point $c_k(t_i)$. See Fig. 5 ① a. The function g consists of the observed velocity magnitude and a function that depends on the distance of point $c_{k+1}(t_i)$ to the center of the vortex lens region for regularization. See Fig. 5 ① b.

In our framework, the proximal algorithm solves for an unknown 5D vector \mathbf{x} , of the position c_k in coordinates (x_k, y_k) , and the reference frame coefficients (a_k, b_k, ω_k) that define the observer field w_k . That is,

$$\mathbf{x} = \begin{bmatrix} x_k & y_k & a_k & b_k & \omega_k \end{bmatrix}^T. \quad (14)$$

The proximal algorithm to solve Eq. 13 above is given by the iterations

$$\begin{aligned} \mathbf{x}^{k+1} &:= \operatorname{prox}_{\lambda f}(\mathbf{z}^k - \mathbf{u}^k), \\ \mathbf{z}^{k+1} &:= \operatorname{prox}_{\lambda g}(\mathbf{x}^{k+1} + \mathbf{u}^k), \\ \mathbf{u}^{k+1} &:= \mathbf{u}^k + \mathbf{x}^{k+1} - \mathbf{z}^{k+1}. \end{aligned} \quad (15)$$

This algorithm uses two auxiliary vectors \mathbf{z} and \mathbf{u} with the same five components as the vector \mathbf{x} . The vectors \mathbf{z} and \mathbf{x} iteratively converge to each other and the optimum, while the vector \mathbf{u} gradually vanishes. The components (x_k, y_k) of the vector \mathbf{z}^k are initialized with the (x, y) coordinates of the start point $c_0(t_i)$, and (a_k, b_k, ω_k) with the (a, b, ω) defining the observer field $w_0(t_i)$. The vector \mathbf{u} is initialized as $\mathbf{u}^k = 0$.

In each iteration, the algorithm first evaluates the proximal operator of the function f (see below), and then evaluates the proximal operator of the function g (also see below). Then the residual vector \mathbf{u} is updated.

The general proximal operator $\operatorname{prox}_{\lambda h}$ of a function h is defined as

$$\operatorname{prox}_{\lambda h}(\mathbf{y}) := \operatorname{argmin}_{\mathbf{x}} \left(h(\mathbf{x}) + \frac{1}{2\lambda} \|\mathbf{x} - \mathbf{y}\|^2 \right). \quad (16)$$

That is, the solution of the proximal operator is itself defined as the solution of a minimization problem. The parameter λ regularizes how far away the new iterate is allowed to be from the previous iterate [31].

Proximal operator of observed time derivative. The proximal operator of the observed time derivative, with the function f below, is

$$\operatorname{prox}_{\lambda f}(\mathbf{y}) := \operatorname{argmin}_{\mathbf{x}} \left(f(\mathbf{x}) + \frac{1}{2\lambda} \|\mathbf{x} - \mathbf{y}\|^2 \right), \quad (17)$$

with the function f defined as (see Eq. 2)

$$f(\mathbf{x}) = \frac{\mu}{A} \int_{R(t_i)} \|L_w(\mathbf{v} - \mathbf{w})(c_k(t_i), t_i)\| dA. \quad (18)$$

In order to be able to evaluate the observed time derivative in the function f above in an iterative fashion, where we do not yet know the observer field w for the time step t_{i+1} , we make the simplifying assumption that, at the time step t_i , the observer field w is a steady velocity field. That is, we conceptually set $\frac{\partial w}{\partial t} = 0$ for the optimization of the observer field in this time step. Although the time derivative does, in general, not vanish for the final optimized field w for all time steps, this simplification allows us to proceed iteratively at interactive computation rates for our optimization problem. Correspondingly, in the world line integration step described below in Sec. 5.3, we make the same simplifying assumption. We can also view this simplification

as being similar to a single stage in a Runge-Kutta integration method, in order to proceed in the time direction. We note that this approach does not imply, however, that the entire obtained observer field \mathbf{w} will be a steady velocity field: It will, in general, be a different vector field $\mathbf{w}(x, t)$ in each time step t , and thus \mathbf{w} will be an unsteady field.

Proximal operator of critical point criterion. The proximal operator of the critical point criterion, with the function $g = g_1 + g_2$ defined below, is given by

$$\text{prox}_{\lambda g}(\mathbf{y}) := \underset{\mathbf{x}}{\operatorname{argmin}} \left(g_1(\mathbf{x}) + g_2(\mathbf{x}) + \frac{1}{2\lambda} \|\mathbf{x} - \mathbf{y}\|^2 \right). \quad (19)$$

The function g_1 for the observed velocity magnitude is (see Eq. 2)

$$g_1(\mathbf{x}) = \|\mathbf{v}(c_k, t_i) - \mathbf{w}(c_k, t_i)\|. \quad (20)$$

The function g_2 is a vortex lens region regularization term given by

$$g_2(\mathbf{x}) = \max(0, \|c_k - c_0\| - r_s). \quad (21)$$

In this regularization term, c_0 is the center of the vortex lens region $R(t_i)$ at time t_i . This regularizer becomes larger than zero when the point c_k is outside of the search radius r_s in the vortex lens region. The vector \mathbf{x} again contains the two coordinates of the position c_k , and the three coefficients that uniquely define \mathbf{w} , to evaluate the functions g_1 and g_2 .

Minimizing the L_1 norm avoids the well-known problem of the L_2 norm that it tends to spread out small errors over the entire domain. While the L_2 norm also can be used in our framework, we have experimentally observed lower error rates (Sec. 6.1) with the L_1 norm.

Also note that in all equations we use physical units (spatial positions and distances, time, velocities), which ensures that the optimization result does not depend on arbitrary scale factors or coordinate systems.

5.2 Stopping Criteria

After the ADMM has converged to a locally optimal reference frame transformation $\mathbf{w}(t_i)$ and a vortex core line position $c(t_i)$, we check if any of the stopping criteria are met. We first check if the angle between $\mathbf{v}(c(t_i), t_i)$ and $\mathbf{w}(c(t_i), t_i)$ is larger than a user-specified tolerance. In addition, we check whether the point is outside the search region $R(t_i)$. If one of the two criteria is fulfilled, we assume that this is the end of the vortex core line candidate's curve and stop. If the criteria pass, we use the solution in the subsequent integration step. See Fig. 5 (2).

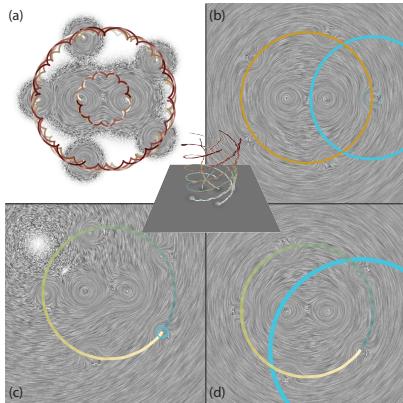


Fig. 6: Seven Rotating Centers: (a) LIC with path lines seeded at the centers of the LIC-patterns. No critical point. (b) Large vortex lens region (blue) centered on one of the outer centers. Critical point matches observed-LIC pattern center. The tracked critical point is shown as orange curve. (c) and (d) Two different-size vortex lenses: (c) Small region: locally the observed LIC pattern matches the critical point. A path line (green-yellow) confirms critical point location. Globally observed LIC patterns do not match. (d) Large region: LIC patterns match globally. Center (space-time with vertical time axis): Seven core lines with several path lines each that closely swirl around the vortex cores. Same for (c) and (d): The core lines are *independent* of the vortex lens size. However, the computed Killing fields resulting from (c),(d) are very different.

5.3 Observer World Line Integration

As depicted in Fig. 5 (2), we integrate the observer field $\mathbf{w}(t_i)$ to compute the next start point $c_0(t_{i+1})$ for the optimization at time t_{i+1} . From the known point $c(t_i)$, resulting from the optimization at time t_i , we numerically integrate a short segment of the world line $t \mapsto w(t)$ through the observer field \mathbf{w} for a given time interval Δt using RK-4 integration and the approach given below. After the integration step, we check if the next point is within the space and time boundaries of the data set. If the new point is outside the time domain or the spatial domain, we stop. Otherwise, we initialize the next optimization step with the current point and observer field and continue with the next time step.

Integration of observer motion. To integrate a world line $t \mapsto w(t)$ of the observer field \mathbf{w} , we choose an integration interval Δt and use Runge-Kutta RK-4 integration. For each RK-4 stage, which is akin to one Euler step, we have the simplifying assumption that the observer field \mathbf{w} is a steady velocity field, i.e., we conceptually set $\frac{\partial \mathbf{w}}{\partial t} = 0$. However, we run alternating ADMM iterations (Sec. 5.1) in each RK-4 stage to update the field $\mathbf{w}(t)$. For this reason, our optimization corresponds to a fully general unsteady observer field \mathbf{w} with $\frac{\partial \mathbf{w}}{\partial t} \neq 0$.

6 EVALUATION

The accuracy of our optimization determines the quality of our observer-relative visualizations, as well as the quality of the extracted core lines that are used for direct visualization and for the advection of the vortex lens through time. To measure the quality of a vortex core line we propose to compute a *Lagrangian Deviation Error Rate* (LDER) as defined below. We use LDER to verify that our approach produces results that converge to a Lagrangian particle trajectory with decreasing integration step size. Further, we use the average of LDER over all extracted vortex core lines to demonstrate that our interactive approach competes with state-of-the-art core line extraction methods. The last part of our evaluation demonstrates that it is possible to find the same set of vortices as with state-of-the-art methods. We show that the set of vortex core lines detected by other methods relies on pre-processing and post-filtering. Since the post-filtering method can produce false positives (i.e., wrongly detected core lines as a result of too little filtering), as well as false negatives (i.e., missed core lines as a result of too aggressive filtering), a manual review of each core line is necessary. Therefore, we argue that an interactive method with tightly coupled detection and reviewing is a beneficial addition to the overall workflow.

6.1 Lagrangian Deviation Error Rate

To evaluate the quality of our core line integration approach, we compute how much the core line deviates from a true Lagrangian solution to

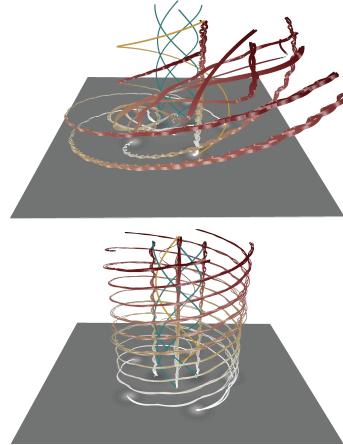


Fig. 7: Dynamic reference frame transformations. Observer-relative path lines are shown for two different reference frame transformations. Top: Relative to the reference frame transformation of one of the small outer vortices. Bottom: Relative to the reference frame transformation of one of the critical points on the inside. The path lines in the center clearly show that only two of the critical points in the center are vortices.

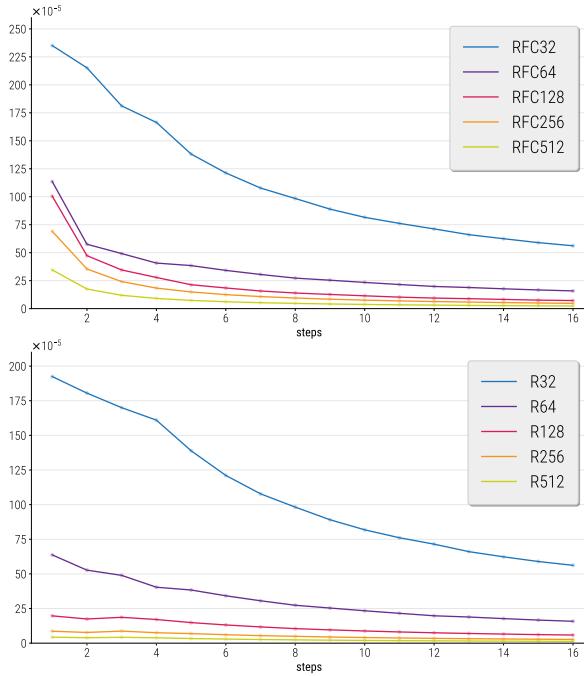


Fig. 8: **Convergence of the average Lagrangian deviation error rate (LDER)** for the rotating four centers data set, for increasing number of integration steps n (horizontal axes) over fixed time interval T . Different resolutions (32 to 512) of (top) input field \mathbf{v} , (bottom) observer field \mathbf{w} .

Table 1: Average Lagrangian deviation error rate comparisons.

	RFC512 (\mathbf{v})	R512 (\mathbf{w})	Cylinder	Boussinesq
affine [15]	0.000120	0.000107	0.000913	0.001579
displacement [2]	0.000245	0.000099	0.001615	0.002036
objective [12]	0.000132	0.000096	0.000946	0.001222
PV [36]	0.024123	0.005378	0.001639	0.001989
similarity [15]	0.000135	0.000097	0.000942	0.001443
vortex lens	0.000345	0.000043	0.000975	0.000478

the ODEs $\dot{\mathbf{v}}(t) = \mathbf{v}(\mathbf{v}(t), t)$, and $\dot{\mathbf{w}}(t) = \mathbf{w}(\mathbf{w}(t), t)$, respectively. Since numerical integration errors are unavoidable, we measure this deviation by integrating path line segments $t \mapsto \mathbf{v}(t)$ and $t \mapsto \mathbf{w}(t)$, in the fields \mathbf{v} and \mathbf{w} , respectively, computing their average deviation from the core line $t \mapsto c(t)$ over a fixed time interval T . We start a path line $t \mapsto u(t)$, with $u := \mathbf{v}$ or $u := \mathbf{w}$, respectively, at a point $c(t_i)$ on the core line at time t_i , and integrate for time $T = t_{i+n} - t_i$, with n integration steps. The average deviation from the Lagrangian trajectory $t \mapsto u(t)$ then is

$$\text{LDER}(c; t_i, T) = \frac{1}{T} \sum_{j=1}^n \|u(t_{i+j}) - c(t_{i+j})\| (t_{i+j} - t_{i+j-1}). \quad (22)$$

In Fig. 8, we analyze the average Lagrangian deviation error rates for all core lines of the rotating four centers (RFC) test data set. For this data set, the optimal reference frame is known, and we can therefore analyze our method with respect to the ground truth. Different grid resolutions (Fig. 8, top: RFC32 to RFC512, input field \mathbf{v} ; Fig. 8, bottom: R32 to R512, observer field \mathbf{w}), and different numbers n of integration steps in the fixed time interval T in Eq. 22 (Fig. 8, horizontal axes), affect the LDER values for both the field \mathbf{v} (Fig. 8, top) and the field \mathbf{w} (Fig. 8, bottom). We can see that the average error rate decreases with decreasing step size (i.e., increasing n for fixed T) and with increasing data resolution. We conclude that our approach is able to find core lines that get closer to Lagrangian particle trajectories with higher sampling rates of the input data as well as smaller step sizes of our integration.

We compare our approach to several state of the art methods. We extract vortex core lines using the parallel vectors (PV) operator [36] as implemented in VTK. Then we compute optimal reference frames¹

¹source provided at <https://github.com/tobguentz/optimal-frames>

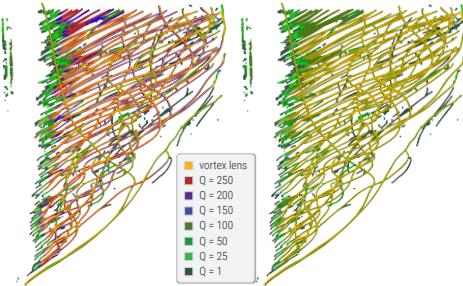


Fig. 9: **Post evaluation of core line candidates.** We use increasingly aggressive thresholds for the Q-criterion and overlay the results for [12]. Left: all filtering results overlaid, additionally our result is overlaid on top. Right: sets of too aggressive Q-thresholds removed. The remaining sets might contain false positives and are subject to manual review.

using objectivity [12], and affine and similarity invariance [15], as well as invariance under displacement transformations [2]. Vortex cores are extracted from these optimal frames by connecting critical points of the transformed vector field [12, Sec. 5.1]. The core lines are subsequently filtered using the Q-criterion to remove noise and critical curves that are not vortex core lines. In Table 1, we show the resulting average error rates of the six methods measured on three test data sets. For the rotating four centers data set (Table 1, RFC512 and R512), we measure Lagrangian deviation error rates (Eq. 22) with respect to the input field (\mathbf{v}) as well as the known optimal observer field (\mathbf{w}). For a fair comparison in Table 1, we use one integration step for each time step for our method. We conclude that our method is comparable and even outperforms other methods in some cases. Reducing the step size of our method decreases the error considerably, as shown in Fig. 8.

6.2 Assessment of Completeness

To assess that the set of vortex core lines that can be found with our method is complete we used the core lines that were extracted with the other five methods and compared the results. We observed that for test data sets with ground truth available the resulting set of core lines is identical. For simulated data the absence of ground truth makes it hard to evaluate for completeness. The other methods rely heavily on post-filtering, and the results typically depend on several filter criteria.

In Fig. 9, we show different filtering results of the set of core lines that were extracted using the objective vortices method [12]. We filter the set of core lines with increasingly aggressive thresholds of the Q-criterion. Core line candidates with higher Q-thresholds are strict subsets of filtering results with lower Q-thresholds. While thresholds above 100 result in false negatives (missing core lines, visible on the left of Fig. 9), Q-thresholds below 100 result in many false positives (critical points that are not core lines visible on the right of Fig. 9). With our method (overlaid yellow in Fig. 9) we obtained a result most similar to a threshold of 50. In the absence of ground truth we do not attempt to argue which method produces better results, but rather want to point out that for a reliable result a careful review of core line candidates with interactive visualization methods is always necessary.

6.3 Parameters for Experiments

The optimization steps that are the solutions to the proximal operators can be implemented with a common solver. We tried several solvers including Gradient Descent, Levenberg-Marquardt, and Gauss-Newton. While these methods compute the same solution they differ in terms of numerical stability and computation times. While Gradient Descent and Levenberg-Marquardt have approximately the same performance, Gauss-Newton was slower by one order of magnitude for our problem. We experimentally found that the λ parameter setting of the proximal ADMM algorithm is not critical, but the algorithm performed best in terms of performance when $\lambda = 100$. The region size parameters of our method have a clear meaning. The parameter r must be set by the user as it depends on the size of the feature that shall be tracked. Although small r yield good results, the user might want to track features with larger reference frames (Sec. 7.1). All results (except for the discussion

of region size) were produced with a region radius that is equal to the cell size of the data grid. The region size r influences the number of Riemann sum terms that are used to estimate the time derivative integral over $R(t)$, and thus directly impacts performance. To test the performance of our approach (and for the region size discussion) we have increased the number of terms. For less than 400 terms our method was fully interactive, providing responsive lens interaction with more than 20 fps. However, for the results shown in this paper we used just five terms. The search region parameter r_s was consistently set to eight times the grid cell size. This gives enough room for the optimization to correct the error that was done in the numerical integration step. The stopping criterion that is concerned with the angle between the vectors $\mathbf{v}(c(t), t)$ and $\mathbf{w}(c(t), t)$ was set to 2 degrees in all our experiments.

7 RESULTS AND DISCUSSION

We demonstrate different aspects of our framework on analytically-defined test cases, and on numerically-simulated flow fields. The supplemental video shows recordings of interactive exploration sessions using our tool. Our vortex lens method runs at interactive frame rates.

7.1 Region Size Dependence

The size of the computational region is an important parameter of recent reference frame computation methods. In the examples in Figs. 6 and 7, we illustrate the influence of the region size on the obtained results. In Fig. 6, we show a synthetic data set that consists of seven vortex cores. Five small vortex regions rotate around two larger ones. In Fig. 6(a), LIC and path lines seeded at the centers of the velocity patterns are shown. Fig. 6(b) shows the extracted vortex core lines of the outer five vortex cores in orange. We show observed LIC with a reference frame computed in the large blue region. The centers of the outer vortices in the observed LIC match the orange core lines. In Fig. 6(c) and (d), we show the effect of the region size on the observed velocities. The small region (blue) in Figure 6(c) captures the local reference frame. A path line (green-yellow circle) shows that the critical point only matches the small vortex in the blue region. In Fig. 6(d), a larger region is used to compute the reference frame transformation. The observed critical points match the core line. The overlay in the center of Fig. 6 shows the extracted seven vortex core lines as well as swirling path lines.

The obtained core lines are *independent* of the region size. However, the result of our optimization is not only a core line but also a reference frame transformation, and this transformation depends on the region size. In Fig. 7, we show a comparison of two reference frame transformations. The top image in Fig. 7 are the observed path lines relative to the observer that was computed for the yellow core line. The small region makes the reference frame transformation adapted to the small vortex only. At the bottom of Fig. 7, the same path lines are shown relative to the observer that was extracted for the yellow curve. A larger region was used to compute a reference frame for the two rotating vortices in the center. This example shows that the extraction of vortex core line candidates is *independent* of the region size. However, the reference frame transformation that is computed to follow the critical point is *not unique* and depends on the region size. Users can therefore choose a region size that matches the feature that they want to visualize relative to the reference. We note that the optimal reference frame depends on the scale of the feature that the user wants to visualize or track over time. A comprehensive analysis of a flow would include the analysis of optimal reference frames on different scales. Although we do not further explore this here, we consider this an important insight for future work on multi-scale feature detection and visualization.

7.2 Exploratory Visualization and Candidate Review

We show example results from short interactive sessions using our interactive tool. The example in Fig. 1 is the visualization of vortex cores from a simulation of a heated cylinder. Small vortices develop and traverse across the simulation domain. In Fig. 4, we show additional close-ups of the same interactive session. Path lines can be seen to closely swirl around the extracted core lines. The color coding on the planes in Fig. 4 visualizes vorticity. The insets show examples of vortex lens regions with observed LIC. In Fig. 10, the result of an interactive

exploration session can be seen. The simulation of a viscous 2D flow around two obstacles starts out as a laminar flow, to later develop vortices behind the second obstacle. The circulating flow behind the corners partially exhibits swirling motion as can be seen from the path lines. Fig. 10 (top) is a visual summary of the data set using core lines and pathlines. The color coding on the time slices is observed vorticity magnitude. At the bottom of Fig. 10, we show the extracted core lines from our result, overlaid on top of the result from five other state of the art methods. We see that the results of the different methods are in consensus behind the second cylinder on the right. The flow behind the first cylinder on the left is mostly laminar with little rotation. This is the region where the results of the methods differ the most.

In Fig. 11 (supplementary), we show the extracted vortex cores of a simulated vortex street behind a cylinder obstacle. The behavior of the flow is easily understood from a few vortex core lines. Towards the end of the spatial simulation domain the vortices become weaker and transition into laminar flow as can be seen from the path lines seeded close to some critical points.

8 CONCLUSIONS

We have presented the first interactive framework that performs fully interactive optimal reference frame optimization for vortex discovery in unsteady 2D fluid flows. Our approach is interactively steered by the user via a vortex lens metaphor that determines the region of interest in which vortex structures are sought via mathematical optimization. Moreover, we also perform the first *joint* optimization of a vortex core line together with the reference frame motion relative to which a vortex core can be detected. This is done interactively using efficient ADMM iterations. Nevertheless, our optimization result only suggests an optimal candidate for a vortex core line. Final evaluation is required to be done by the user. For this purpose, we provide a variety of visualization and interaction techniques, including real time transformation of the unsteady input flow field into the interactively computed reference frame, observed LIC in a lens, and lens advection through time. Overall, we have shown that the combination of interaction and joint optimization allows the user to obtain insights using observed flow visualization as well as high-quality vortex core line structures that are guaranteed to be close to Lagrangian particle trajectories in the original input flow field.

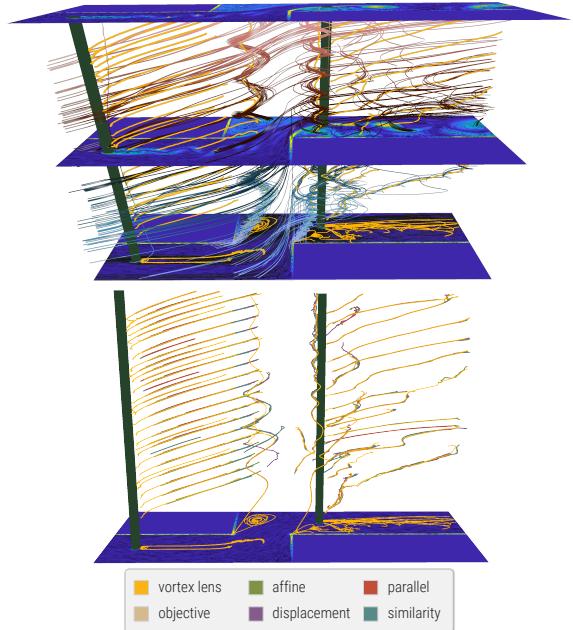


Fig. 10: **Simulated viscous flow** around a corner and two obstacles: Space-time visualization (time is vertical). Top: result of an interactive vortex candidate extraction session. Vortex core candidates (yellow) and path lines seeded close to critical points. Path lines can be used to distinguish core lines from other critical points. Bottom: Extracted vortex core lines (ours in yellow) overlaid with results from other approaches.

ACKNOWLEDGMENTS

We thank Anna Frühstück for help with the figures and the video. This work was supported by King Abdullah University of Science and Technology (KAUST). This research used resources of the Core Labs of King Abdullah University of Science and Technology.

REFERENCES

- [1] G. Astarita. Objective and generally applicable criteria for flow classification. *Journal of Non-Newtonian Fluid Mechanics*, 6(1):69–76, 1979. doi: 10.1016/0377-0257(79)87004-4 3
- [2] 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 1, 3, 8
- [3] D. C. Banks and B. A. Singer. A predictor-corrector technique for visualizing unsteady flow. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):151–163, 1995. doi: 10.1109/2945.468404 3
- [4] D. Bauer and R. Peikert. Vortex Tracking in Scale-Space. In *Proceedings of Eurographics/IEEE TCVG Symposium on Visualization 2002*, pp. 233–241, 2002. doi: 10.2312/VisSym/VisSym02/233-240 3
- [5] M. Berenjebkoub, G. Chen, and T. Günther. Vortex boundary identification using convolutional neural network. In *IEEE VIS 2020 Short Papers*, pp. 261–265, 2020. doi: 10.1109/VIS47514.2020.00059 3
- [6] 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 3
- [7] 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 3
- [8] 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 3
- [9] B. Cabral and L. C. Leedom. Imaging vector fields using line integral convolution. In *Proceedings of SIGGRAPH 1993*, pp. 263–270, 1993. doi: 10.1145/166117.166151 3
- [10] R. Drouot and M. Lucius. Approximation du second ordre de la loi de comportement des fluides simples. Lois classiques déduites de l'introduction d'un nouveau tenseur objectif. *Archiwum Mechaniki Stosowanej*, 28(2):189–198, 1976. 3
- [11] A. Globus, C. Levit, and T. Lasinski. A tool for visualizing the topology of three-dimensional vector fields. In *Proceedings of IEEE Visualization 1991*, pp. 33–40, 1991. doi: 10.1109/VISUAL.1991.175773 3
- [12] 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 1, 2, 3, 8
- [13] T. Günther, M. Schulze, and H. Theisel. Rotation invariant vortices for flow visualization. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):817–826, 2016. doi: 10.1109/tvcg.2015.2467200 3
- [14] 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, 3
- [15] T. Günther and H. Theisel. Hyper-objective vortices. *IEEE Transactions on Visualization and Computer Graphics*, 26(3):1532–1547, 2020. doi: 10.1109/TVCG.2018.2868760 8
- [16] 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 1, 2, 3, 5
- [17] G. Haller. An objective definition of vortex. *Journal of Fluid Mechanics*, 525:1–26, 2005. doi: doi:10.1017/S0022112004002526 1, 3
- [18] G. Haller. Can vortex criteria be objectivized? *Journal of Fluid Mechanics*, 508:A25, 2003. doi: 10.1017/jfm.2003.1715 3
- [19] 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 3
- [20] G. A. Holzapfel. *Nonlinear Solid Mechanics: A Continuum Approach for Engineering*. Wiley, 2000. 3
- [21] 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. 3
- [22] J. Jeong and F. Hussain. On the identification of a vortex. *Journal of Fluid Mechanics*, 285:69–94, 1995. doi: 10.1017/S0022112095000462 3, 4
- [23] B. Jobard, G. Erlebacher, and M. Y. Hussaini. Lagrangian-Eulerian advection of noise and dye textures for unsteady flow visualization. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):211–222, 2002. doi: 10.1109/TVCG.2002.1021575 3
- [24] B. Jobard and W. Lefer. Creating evenly-spaced streamlines of arbitrary density. In *Eurographics Workshop on Visualization in Scientific Computing 1997*, pp. 211–222, 1997. doi: 10.1007/978-3-7091-6876-9_5 3
- [25] B. Kim and T. Günther. Robust reference frame extraction from unsteady 2d vector fields with convolutional neural networks. *Computer Graphics Forum*, 38(3):285–295, 2019. doi: 10.1111/cgf.13689 3
- [26] R. S. Laramee, B. Jobard, and H. Hauser. Image space based visualization of unsteady flow on surfaces. In *Proceedings of IEEE Visualization 2003*, pp. 123–130, 2003. doi: 10.1109/visual.2003.1250364 3
- [27] G.-S. Li, X. Tricoche, and C. Hansen. GPUFLIC: Interactive and accurate dense visualization of unsteady flows. In *Proceedings of Eurographics/IEEE VGTC Symposium on Visualization 2006*, pp. 29–34, 2006. doi: 10.2312/VisSym/EuroVis06/029-034 3
- [28] H. J. Lugt. The dilemma of defining a vortex. In *Recent Developments in Theoretical and Experimental Fluid Mechanics: Compressible and Incompressible Flows*, pp. 309–321. Springer-Verlag, 1979. doi: 10.1007/978-3-642-67220-0_32 1
- [29] R. W. Ogden. *Non-Linear Elastic Deformations*. Dover Publications, Inc., 1997. 3
- [30] A. Okubo. Horizontal dispersion of floatable particles in the vicinity of velocity singularities such as convergences. *Deep Sea Research and Oceanographic Abstracts*, 17(3):445–454, June 1970. doi: 10.1016/0011-7471(70)90059-8 3
- [31] N. Parikh and S. Boyd. *Proximal Algorithms*. Now Publishers Inc., 2013. doi: 10.1561/2400000003 2, 6
- [32] 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 3
- [33] A. E. Perry and M. S. Chong. A description of eddying motions and flow patterns using critical-point concepts. *Ann. Rev. Fluid Mech.*, 19(1):125–155, 1987. doi: 10.1146/annurev.fl.19.010187.001013 3
- [34] A. E. Perry and M. S. Chong. Topology of flow patterns in vortex motions and turbulence. *Applied Scientific Research*, 53(3):357–374, 1994. doi: 10.1007/bf00849110 3
- [35] 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 1, 2, 3, 5
- [36] M. Roth and R. Peikert. A higher-order method for finding vortex core lines. In *Proceedings of IEEE Visualization 1998*, pp. 143–150, 1998. doi: 10.1109/visual.1998.745296 8
- [37] F. Sadlo, R. Peikert, and E. Parkinson. Vorticity based flow analysis and visualization for pelton turbine design optimization. In *Proceedings of IEEE Visualization 2004*, pp. 179–186, 2004. doi: 10.1109/visual.2004.128 3
- [38] 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 3
- [39] H.-W. Shen and D. L. Kao. UFLIC: A line integral convolution algorithm for visualizing unsteady flows. In *Proceedings of IEEE Visualization 1997*, pp. 317–322, 1997. doi: 10.1109/visual.1997.663898 3
- [40] D. Stalling and H.-C. Hege. Fast and resolution independent line integral convolution. In *Proceedings of SIGGRAPH 1995*, pp. 249–256, 1995. doi: 10.1145/218380.218448 3
- [41] 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 3
- [42] H. Theisel, M. Hadwiger, P. Rautek, T. Theußl, and T. Günther. Vortex criteria can be objectivized by unsteadiness minimization. *Physics of Fluids*, 33(10):107115, 2021. doi: 10.1063/5.0063817 3
- [43] H. Theisel, J. Sahner, T. Weinkauf, H.-C. Hege, and H.-P. Seidel. Extraction of parallel vector surfaces in 3d time-dependent fields and application to vortex core line tracking. In *Proceedings of IEEE Visualization 2005*, pp. 631–638, 2005. doi: 10.1109/visual.2005.1532851 3
- [44] H. Theisel and H.-P. Seidel. Feature flow fields. In *Proceedings of*

- Eurographics/IEEE TCVG Symposium on Visualization 2003*, pp. 141–148, 2003. doi: [10.2312/VisSym/VisSym03/141-148](https://doi.org/10.2312/VisSym/VisSym03/141-148) 2, 3
- [45] X. Tricoche, T. Wischgoll, G. Scheuermann, and H. Hagen. Topology tracking for the visualization of time-dependent two-dimensional flows. *Computers & Graphics*, 26(2):249–257, 2002. doi: [10.1016/s0097-8493\(02\)00056-0](https://doi.org/10.1016/s0097-8493(02)00056-0) 3
 - [46] C. Truesdell and W. Noll. *The Nonlinear Field Theories of Mechanics*. Springer-Verlag, 1965. doi: [10.1007/978-3-662-10388-3_1](https://doi.org/10.1007/978-3-662-10388-3_1) 1, 3
 - [47] J. J. van Wijk. Image based flow visualization. *ACM Transactions on Graphics*, 21(3):745–754, 2002. doi: [10.1145/566570.566646](https://doi.org/10.1145/566570.566646) 3
 - [48] J. J. van Wijk. Image based flow visualization for curved surfaces. In *Proceedings of IEEE Visualization 2003*, pp. 123–130, 2003. doi: [10.1109/visual.2003.1250363](https://doi.org/10.1109/visual.2003.1250363) 3
 - [49] T. Weinkauf, J. Sahner, H. Theisel, and H.-C. Hege. Cores of swirling particle motion in unsteady flows. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1759–1766, 2007. doi: [10.1109/tvcg.2007.70545](https://doi.org/10.1109/tvcg.2007.70545) 3
 - [50] 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](https://doi.org/10.1016/0167-2789(91)90088-q) 3
 - [51] A. Wiebel, R. Chan, C. Wolf, A. Robitzki, A. Stevens, and G. Scheuermann. Topological flow structures in a mathematical model for rotation-mediated cell aggregation. In *Topological Methods in Data Analysis and Visualization, Mathematics and Visualization*, pp. 193–204, 2009. doi: [10.1007/978-3-642-15014-2_16](https://doi.org/10.1007/978-3-642-15014-2_16) 3
 - [52] C. Xie, L. Xing, C. Liu, and X. Li. Multi-scale vortex extraction of ocean flow. In M. L. Huang, ed., *Visual Information Communication*, pp. 173–183. Springer-Verlag, 2010. doi: [10.1007/978-1-4419-0312-9_11](https://doi.org/10.1007/978-1-4419-0312-9_11) 3
 - [53] 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](https://doi.org/10.1109/tvcg.2021.3115565) 2, 3, 4, 5

A SUPPLEMENTARY RESULTS

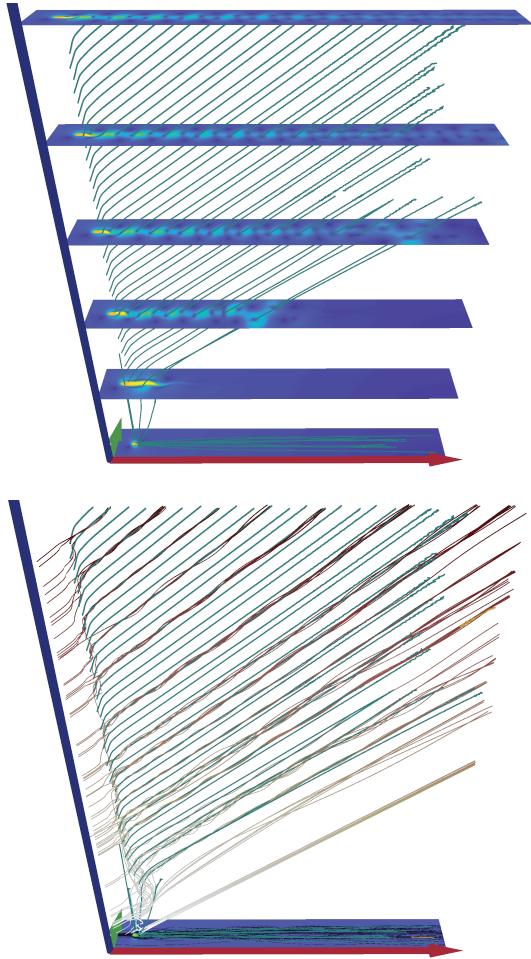


Fig. 11: **Vortex street** behind a cylinder obstacle (space-time visualization): Top: critical curves (green) extracted using the vortex lens. The planes show observed velocities (blue-yellow color coding). Bottom: Path lines are manually seeded close to critical points to verify that the extracted curves are vortex cores.