An Interactive Exploration System for Physically-Observable Objective Vortices in Unsteady 2D Flow

Thesis by

Xingdi Zhang

In Partial Fulfillment of the Requirements

For the Degree of

Masters of Science

King Abdullah University of Science and Technology

Thuwal, Kingdom of Saudi Arabia

September, 2021

EXAMINATION COMMITTEE PAGE

The thesis of Xingdi Zhang is approved by the examination committee

Committee Chairperson: Markus Hadwiger
Committee Members: Helmut Pottmann, Ivan Viola

ABSTRACT

An Interactive Exploration System for Physically-Observable Objective Vortices in Unsteady 2D Flow

Xingdi Zhang

Vortex detection has been a long-standing and challenging topic in fluid analysis. Recent state-of-the-art extraction and visualization of vortices in unsteady fluid flow employ objective vortex criteria, which makes feature extraction independent of reference frames or observers. However, even objectivity can only guarantee that different observers reach the same conclusions, but not necessarily guarantee that these conclusions are the only physically meaningful or relevant ones. Moreover, a significant challenge is that a single observer is often not sufficient to accurately observe multiple vortices that follow different motions.

This thesis presents a novel mathematical framework that represents physically realizable observers as the Lie algebra of the Killing fields on the underlying manifold, together with a software system that enables the exploration and use of an interactively chosen set of observers, resulting in relative velocity fields and objective vortex structures in real-time. Based on our mathematical framework, our system facilitates the objective detection and visualization of vortices relative to well-adapted reference frame motions, while at the same time guaranteeing that these observers are physically realizable.

We show how our framework speeds up the exploration of objective vortices in unsteady 2D flow, on planar as well as on spherical domains.

ACKNOWLEDGEMENTS

This thesis was completed under the careful guidance and help of my two supervisors, professor Markus Hadwiger and research scientist Peter Rautek. I am grateful for their kindness and support in my study and life. I also thank Anna Fruhstück for her help with the figures and the video. This research used resources of the Core Labs of King Abdullah University of Science and Technology. Finally, I also thank my mom and my girl friend Mingxia Li for their company.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

1.1 Motivation

Flow visualization is a fundamental research area that has been an indispensable tool in fluid dynamics research and has been employed extensively in numerous areas including aerodynamics, meteorology, oceanography, continuum mechanics, computer simulation, etc [4, 5]. It aims at providing intuitive visualization and feature extraction for fluid flow, and it acts as a valuable and human-friendly tool for studying complex flow structures. Visualization results facilitate qualitative assessments that guide the researcher in the analysis.

One of the most important research topics in flow visualization is vortex extraction, which aims to find vortex structure in a flow field and has been a long-standing unsolved problem. To extract and visualize features in fluid flow, it often depends on the reference frames (observers), relative to which the input velocity field is given. Simply speaking, an observer can be thought of as a camera that records velocities relative to its own motion.

A large amount of recent research in flow visualization has built on the concept of objectivity [6, 7, 8, 3]. Objectivity is a property that the conclusions are independent of the choice of observers. Different observers will agree on one conclusion. For example, in a vector field, although quantities like velocities are relative to the reference frame, i.e., not objective, the locations of vortices are independent of the reference frames.

However, even objectivity, by itself, only guarantees that different observers reach the same conclusions, but not necessarily guarantees that these conclusions are physically meaningful. For many real-world flow phenomena, it is not sufficient to explore a fluid field using just one observer. For example, when vortices move in different directions, observers that travel with them will see different phenomena relative to themselves. In this case, it is obvious that no single observer can be used to depict all features of interest.

As a result, recent research proposed to detect vortices using a large number of observers jointly. The drawback of these methods is that the physical meaningfulness is disputable. In contrast, this thesis deals with an interactive approach that allows to quickly change and modify observers while visualizing the vector field and flow features relative to one observer at a time, as shown in Fig 1.1(a).

Our mathematical framework models all observers as the elements of the Lie algebra of physically-realizable observer motions on the underlying manifold M. This allows us to define a vector space of observers and to efficiently carry out vector space operations like interpolation and averaging. The interactive computation of observer-relative visualizations allows us to quickly explore the space of potentially meaningful observers.

1.2 Problem Statement and Objectives

A fundamental challenge in vortex extraction is to find a suitable reference frame, as the perception of motion is always relative to an observer or reference frame. The visualization of features of the flow field including vortices relies heavily on the choice of the reference frame.

Our objective is to design a novel system that enables users to interactively explore the space of all potentially meaningful reference frames, i.e., physically-realizable observers, while jointly computing and visualizing the resulting vortex structures.

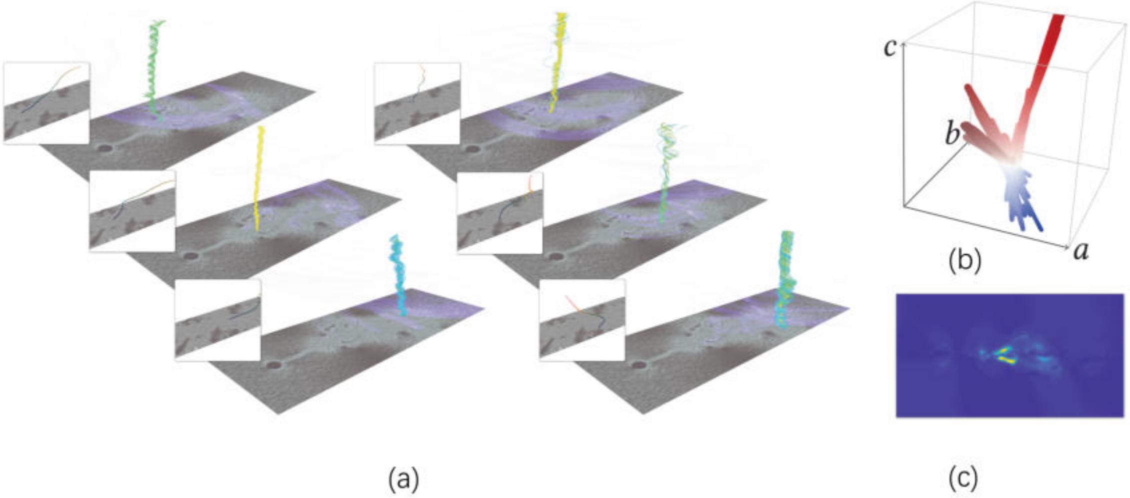


Figure 1.1: **Boussinesq.** In sub-figure (a), there are six observer-relative visualizations. (Every small figure on the top left corner shows the motion of a rigid-body observer and every middle size figure is the corresponding visualization relative to that observer. Vertical axis is time. The path lines are variance-based color-coded.) (b) shows the (a, b, c) parameter space of candidate observers. (c) shows the color-coded observer similarity on the manifold.

Our contribution is summarized as follows:

1. We propose a novel mathematical framework that enables the efficient manipulation of observers by expressing all possible observer motions in the same basis of vector fields for the corresponding Lie algebra.
2. We propose a novel visualization framework that enables the interactive selection, comparison, and user evaluation of objective vortex structures relative to an interactively-chosen and -modified set of physically-realizable observers.
3. We describe how previous methods integrate with our interactive framework and leverage the Lie algebra of observer motions to treat flow fields in 2D flat space as well as on the curved surface of a sphere in a conceptually unified way.

1.3 Thesis Organization

The thesis is organized as follows:

- Chapter 2 covers fundamentals and basic mathematical concepts relevant to this work, including vector calculus, vector fields, differential geometry, the velocity gradient tensor on a differentiable manifold, Lie Groups and Lie algebras, and reference frame invariances.

- Chapter 3 gives a brief overview of previous work that is related to this thesis, including flow visualization, reference frames, vortex extraction, objectivity, Killing vector field.

- Chapter 4 provides a detailed description of our novel mathematical framework, including the explanation of physically-realizable observer, the mathematics of obtaining an observer-relative transformation and the corresponding observed differential quantities.

- In Chapter 5, the design and implementation of the interactive system are carefully described. We describe the basic functionalities of the tool and highlight the novel interactive features. We further analyze the performance of our system in this chapter.

- In Chapter 6, we demonstrate the usage of our interactive system with various flow field data sets, both from analytical settings and numerical simulations, and show how such an interactive exploration tool can be of value in the investigation of vortex structures.

- In Chapter 7, conclusions of this thesis are made, as well as remarks regarding promising future research directions.

Chapter 2

Fundamentals of Flow Visualization

2.1 Notation

In this thesis, I will use italic lowercase letters such as s to denoted scalar quantites. Vector quantities are denoted by lowercase letters with arrow such as $\vec{v}$. Matrices and second-order tensors will be denoted by bold uppercase letters such as $\mathbf{M}$. Following convention, the n-dimensional real number space will be denoted as $\mathbb{R}^n$, the sphere manifold will will be denoted by $\mathbb{S}^2$. For a set or an arbitrary manifold, I will use normal uppercase letters like S, M. $\left\langle \vec{a}, \vec{b} \right\rangle$ denotes the inner product of vector $\vec{a}$ and $\vec{b}$, and × is used for the cross product, and $\otimes$ denotes the tensor product.

2.2 Vector Calculus in Cartesian Coordinates

In this thesis, we need some concepts from vector calculus. This section will review the fundamentals of vector calculus for readers who are not familiar with it.

If we are working in 3D Euclidean space with Cartesian coordinates, the nabla operator ∇ formally denotes the spatial partial derivative symbols

$$\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right)^T . \tag{2.1}$$

One common use of the ∇ operator is in a formal dot product with a two-dimensional or three-dimensional vector field $\vec{v}$. In Cartesian coordinates, this gives the **diver-**

gence of a 2D vector field $\vec{v}$, given by Cartesian components (u, v), as

$$\nabla \cdot \vec{v} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}, \tag{2.2}$$

and the divergence of a 3D vector field $\vec{v}$, given by Cartesian components (u, v, w), as

$$\nabla \cdot \vec{v} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}. \tag{2.3}$$

Another common use of the ∇ operator is in a formal cross product with a three-dimensional vector field $\vec{v}$. This gives the **curl** or so-called **vorticity**. Vorticity corresponds to the local tendency for a vector field to swirl (circulate) around a given position, which is often used for vortex detection. For a 3D vector field $\vec{v}$, given by Cartesian components (u, v, w), the curl is given by

$$\vec{\omega} = \nabla \times \vec{v} = \begin{pmatrix} \frac{\partial w}{\partial y} - \frac{\partial v}{\partial z} \\ \frac{\partial u}{\partial z} - \frac{\partial w}{\partial x} \\ \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \end{pmatrix}. \tag{2.4}$$

The ∇ operator is also often applied to a scalar function f, giving its gradient as

$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right)^T. \tag{2.5}$$

Applied to a velocity vector field, the ∇ operator gives the **velocity gradient tensor**. In Cartesian coordinates, the velocity gradient of a 2D vector field $\vec{v}$ is given as:

$$\nabla \vec{v}(x, y) = \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{pmatrix}. \tag{2.6}$$

In Cartesian coordinates, the velocity gradient of a 3D vector field $\vec{v}$ is given as:

$$\nabla\vec{v}(x,y,z) = \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial x} & \frac{\partial w}{\partial y} & \frac{\partial w}{\partial z} \end{pmatrix}. \tag{2.7}$$

The velocity gradient tensor is the first-order derivative that describes how the flow velocity changes locally at a given position along different directions in space. It plays a very important role in vector field analysis. For example, the velocity gradient tensor can be decomposed into two parts $\nabla\vec{v} = \mathbf{S} + \mathbf{\Omega}$, with:

$$\mathbf{S} = \frac{\nabla\vec{v} + (\nabla\vec{v})^T}{2}, \mathbf{\Omega} = \frac{\nabla\vec{v} - (\nabla\vec{v})^T}{2}. \tag{2.8}$$

Here, $\mathbf{S}$ is the symmetric rate-of-strain tensor, and $\mathbf{\Omega}$ is the anti-symmetric spin tensor, which satisfies the following relationship:

$$\mathbf{\Omega} \cdot \vec{e} = -\frac{1}{2} \cdot (\vec{\omega} \times \vec{e}), \tag{2.9}$$

where $\vec{e}$ is an arbitrary vector, and $\vec{\omega}$ is the curl.

2.3 Steady and Unsteady Vector Fields

A vector field that does not change with time is called a **time-independent vector field** or **steady vector field**. Given a basis, a steady vector field can be viewed as a function $f\colon D \to \mathbb{R}^2$ or $\mathbb{R}^3$, where D is a manifold (e.g., a subset of $\mathbb{R}^2$ or $\mathbb{R}^3$).

If a vector field varies with time, then it is called a **time-dependent vector field** or **unsteady vector field**. That is, at the same spatial position $x \in D$, the vector field $\vec{v}(x,t)$ can have different values at different times $t \in T$. Given a basis, a time-dependent vector field can be viewed as a function $f\colon D \times T \to \mathbb{R}^2$ or $\mathbb{R}^3$.

In this work, we restrict our study to two-dimensional unsteady vector fields on the plane $D = \mathbb{R}^2$ and on the sphere $D = \mathbb{S}^2$. Most mathematical formulations that are presented generalize to three dimensions. For brevity, we will denote an unsteady vector field by $\vec{v}(x, t)$ in the rest of the thesis.

2.4 Stream Lines and Path Lines

Stream objects are a family of visualization techniques that show the trajectory of a certain object (e.g., a particle, or a curve) in a vector field over a given time interval [9]. In our framework, we use two stream objects: streamlines and path lines.

Streamlines. For a steady field or a fixed point in time of an unsteady field, the integral curve that describes the path of a particle released at position x_0 is defined as a streamline. It is given by the solution to the autonomous ODE:

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}t} x(t) = \vec{v}(x(t)), \\ &x(0) = x_0. \end{aligned} \tag{2.10}$$

We will use streamlines to compute line integral convolution (LIC) textures [10].

Path lines. For an unsteady field, the integral curve that describes the path of a particle released at position x_0 at time t_0 is defined as a path line, and is given by the solution to the initial value problem:

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}t} x(t) = \vec{v}(x(t), t), \\ &x(t_0) = x_0. \end{aligned} \tag{2.11}$$

2.5 Differential Geometry

Our framework works on Euclidean spaces as well as curved spaces (e.g., a sphere). To extend the mathematical concepts from the vector calculus in Euclidean space to curved surfaces, it is necessary to apply concepts of differential geometry.

2.5.1 Homeomorphism, Diffeomorphism

Definition 1. *If a function $f : X \to Y$ between two topological spaces has the following properties:*

- *f is a bijection.*
- *f is continuous.*
- *The inverse function f^{-1} is continuous.*

Then we say f is a homeomorphism or bicontinuous function.

If the function f and its inverse f^{-1} are not only continuous but also differentiable then we say f corresponds to a diffeomorphism.

2.5.2 Regular Curve and Tangent Vector

Definition 2. *A parametrized differentiable curve is a differentiable map $\alpha : I \to \mathbb{R}^3$ of an open interval $I = (a, b) \subset \mathbb{R}$ into $\mathbb{R}^3$.*

The definition of parametrized differentiable curve guarantees that the first order derivative $\dot{\alpha}(t)$ of map α exists [11]. There is a well-defined straight line, which contains the point $\alpha(t)$ and the vector $\dot{\alpha}(t)$, called tangent line to α at t. And the vector $\dot{\alpha}(t)$ is called the **tangent vector** or **velocity vector** of curve α.

When a curve is **arc-length parametrized**, the norm of the curve's tangent vector is always one, i.e., $|\dot{\alpha}(t)| = 1$, for all t. Otherwise, the curve's tangent vectors will be scaled by a coefficient and can be represented as $\vec{t} = k * \vec{t_e}, k = |\dot{\alpha}(t)|$, where $\vec{t_e}$ is the unit vector in the direction of $\dot{\alpha}(t)$. Fig. 2.1 shows a tangent vector of curve $\alpha(t)$.

If the tangent vectors of a parametrized differentiable curve $\alpha(t)$ never equal to zero, i.e., $\dot{\alpha}(t) \neq 0$ for all t, this curve is said to be a regular curve. Only a regular curve can guarantee that we always have tangent vectors along the curve.

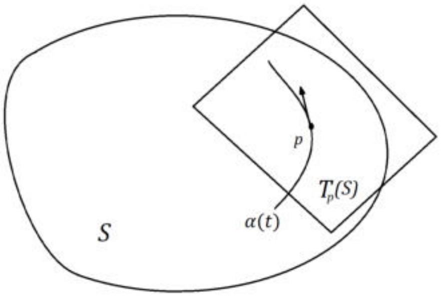


Figure 2.1: **Tangent plane.** Let point $p \in$ Regular Surface S, the velocity vector (tangent vector) of any smooth curve $\alpha(t)$ passing through p lies in the tangent plane of S at point p ($T_p(S)$).

2.5.3 Regular Surface and Manifold

Regular Surface

Do Carmo [11] defines a regular surface as a subset of $S \subset \mathbb{R}^3$ such that for each $p \in S$, there exists a neighborhood V and a map $x : U \to V \cap S$ of an open set $U \subset \mathbb{R}^2$ onto $V \cap S \subset \mathbb{R}^3$ such that:

- x is differentiable.
- x is a homeomorphism.
- For each $q \in U$, the differential $dx_q : \mathbb{R}^2 \to \mathbb{R}^3$ is one-to-one.

If the map x is also infinitely differentiable, i.e ., $x \in C^{(n)}$, then the regular surface is a **smooth surface**.

Manifold

In mathematics, the definition of a manifold is the following:

Definition 3. *An n-dimensional real manifold M is a topological space that each point of M has a neighborhood that is homeomorphic to an open subset of n-dimensional Euclidean space.*

Thus, a manifold is a topological space that locally resembles Euclidean space near each point, which implies that we can define several local coordinate spaces to cover the whole manifold. Every local coordinate space is called a **coordinate chart** (a parametrization or a system of local coordinates). A set of coordinate charts that covers the manifold forms an **atlas**.

Examples of manifold: a regular curve is a 1-dimensional manifold, a regular surface like a sphere or a plane is a 2-dimensional manifold. If a surface is regular then it is also a differentiable manifold [11].

In our framework, we are working on the sphere $\mathbb{S}^2$ and the plane $\mathbb{R}^2$. Both are differentiable manifolds. The plane is a homeomorphism to $\mathbb{R}^2$, and thus can be covered by a single chart. The sphere $\mathbb{S}^2$ is not a homeomorphism to $\mathbb{R}^2$, and we will use six coordinate charts to cover it (See section 5.2.1).

2.5.4 Tangent Space

For a regular surface S, at any point $p \in S$, we have infinitely many regular curves that pass through this point. For any curve that passes through point p, its first derivative corresponds to its tangent vector. Given two different curves pass through point p, the two tangent vectors span a plane, which is called the **tangent plane** (tangent space) of surface S at point p, denoted by $T_p(S)$. The general definition of tangent space is the following:

Definition 4. *Let manifold $M \subseteq \mathbb{R}^n$, the tangent space $T_p(M)$ at $p \in M$ is the set of all tangent vectors $\vec{v} = \dot{\alpha}(t)$, where $\alpha : I \to M$ is a smooth curve with $\alpha(0) = p$; here I is an open interval around zero.*

Fig. 2.1 shows a tangent plane and a tangent vector on a smooth surface. In fact, if a surface is regular, it is guaranteed that there will always be one and only one tangent plane for every point $p \in S$. The tangent planes on the surface S, are called the tangent bundle of S. Tangent spaces are a fundamental concept to define a metric at any point $p \in S$.

2.5.5 Metric Tensor

At any point p of a regular surface $S\colon (u, v) \mapsto x(u, v)$, and given a tangent vector $\vec{w} \in T_p(S)$ (Fig. 2.1), we can define a quadratic form $I_p\colon T_p(S) \to \mathbb{R}$ by

$$I_p(\vec{w}) = \langle \vec{w}, \vec{w} \rangle = \vec{w}^T \mathbf{G}\, \vec{w}, \tag{2.12}$$

where we can write the symmetric tensor $\mathbf{G}$ in components as

$$\mathbf{G} = \begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix} = \begin{pmatrix} x_u \cdot x_u & x_u \cdot x_v \\ x_v \cdot x_u & x_v \cdot x_v \end{pmatrix}. \tag{2.13}$$

Here, x_u, x_v denote the tangent vectors given by the first-order partial derivatives of the map x, i.e., $x_u := \frac{\partial x(u,v)}{\partial u}$, $x_v := \frac{\partial x(u,v)}{\partial v}$, and $\cdot$ denotes the Euclidean dot product.

Arbitrary tangent vectors $\vec{w} \in T_p(S)$ can then be given in components referred to the coordinate basis vectors x_u, x_v in each tangent space $T_p(S)$.

This quadratic form is the **first fundamental form** [11], and the tensor $\mathbf{G}$ is called the **metric tensor**. Thus, the metric tensor determines the inner product of vectors at any point on the surface. It allows us to make measurements on the surface, such as lengths of curves, angles of tangent vectors, and areas of regions.

If we define a positive-definite metric on a differentiable manifold, this manifold is a **Riemannian manifold**. On a Riemannian manifold M, every position p has a metric tensor, and the metric tensors are possibly different for different p.

2.6 General Velocity Gradient Tensors

The vector calculus equations listed in section 2.2 are only correct in Cartesian coordinates and Euclidean spaces. A vector can be represented as a linear combination of components and basis vectors induced by local chart. For instance, a 3-dimensional vector in Cartesian coordinates $\vec{v} = (a, b, c)$ can be represented as $\vec{v} = a\vec{e_1} + b\vec{e_2} + c\vec{e_3}$, where $\vec{e_1}, \vec{e_2}, \vec{e_3}$ are the Cartesian coordinate basis vectors. Because the basis vectors are the same everywhere in Cartesian coordinates, usually the basis vectors are omitted. In other coordinates as well as curved manifolds the basis vectors are different at each point.

2.6.1 Einstein Summation Convention

To define tensors we employ the Einstein Summation Convention [12], which implies summation over indices occurring twice (once "upstairs" and once "downstairs"). For example, to represent a vector $\vec{v}$ referred to a basis $\vec{e_i}$, we pair components v^i with the basis vectors via

$$\vec{v} = v^i\vec{e_i} := \sum_{i=1}^{n} v^i\vec{e_i} = v^1\vec{e_1} + v^2\vec{e_2} + ... + v^n\vec{e_n}.$$

Carrying out the implied summation is called a **contraction**. For an n-dimensional manifold, every paired ("dummy") index represents a sum over all n dimensions.

2.6.2 Vector Bases and Dual Vector Bases

Vector Bases. On any point x of a manifold, we are given a local coordinate system around x. To refer a vector $\vec{v}$ to the local coordinate basis, we expand it in components as $\vec{v} = v^i\vec{e_i}$, where $\vec{e_i}$ is a basis for tangent space T_xM. In general, the tangent space T_xM at different $x \in M$ have different bases.

Dual Vector Bases. The dual basis $\{\omega^i\}$ acts as a selection function for vector components, where $\omega^i(\vec{e}_j) = \delta^i_j$, with Kronecker delta $\delta^i_j = 1$ if $i = j$, and zero otherwise. Each dual base vector ω^i is a covector that linearly maps a vector to a scalar. The dual basis $\{\omega^i\}$ reads off the components of a vector $\vec{v}$ referred to $\vec{e}_i$. $\vec{v} = \omega^i(\vec{v})\vec{e}_i$.

2.6.3 Types of Tensors

We view a tensor field as a multi-linear coordinate-independent map that, at any point $x \in M$, maps a set of vectors and covectors (1-forms) to a scalar. We list the following types of tensors that are used in this thesis [3]. First order tensors and second order tensors can be categorized based on their basis vector types:

- A $\binom{1}{0}$ tensor is a vector referring to a basis $\{\vec{e}_i\}$, and expanded as Einstein form: $v^i\vec{e}_i$.

- A $\binom{0}{1}$ tensor is a 1-form (covector) referring to a dual basis $\{\omega^i\}$, and expanded as $v_i\omega^i$. We can also write $v_i dx^i$ for coordinate bases $\omega^i := dx^i$.

- A $\binom{1}{1}$ tensor T, as a bi-linear map of one covector and one vector argument to a scalar, is referred to a basis $\{\vec{e}_i \otimes \omega^j\}$, and expanded as $T^i_j\vec{e}_i \otimes \omega^j$.

- A covariant second-order tensor (a $\binom{0}{2}$ tensor), such as the metric tensor G, has basis $\{\omega^i \otimes \omega^j\}$, and is expanded to $G_{ij}\omega^i \otimes \omega^j$.

- A contravariant second-order tensor T, (a $\binom{2}{0}$ tensor), has basis $\{\vec{e}_i \otimes \vec{e}_j\}$, and is expanded as $T^{ij}\vec{e}_i \otimes \vec{e}_j$.

2.6.4 Intrinsic Covariant Derivatives

For any vector field $\vec{v}$ given on a manifold M, the velocity gradient tensor $\nabla\vec{v}$ is a second-order tensor field on M that allows computing directional derivatives of the vector field $\vec{v}$ in any direction [3]. In the Cartesian coordinates, we can simply compute

$\nabla\vec{v}$ using equation 2.7. However, for a curved manifold M, the tensor $\nabla\vec{v}$ must be computed intrinsically as the **covariant derivative** of $\vec{v}$.

Definition 5. *Given a vector field $\vec{v}$ on a manifold M, at any $x \in M$, the covariant derivative $(\nabla\vec{v})$ is a second-order tensor field, which maps an arbitrary vector $\vec{x} \in T_xM$ to the directional derivative of the field $\vec{v}$ in the direction $\vec{x}$, i.e,*

$$(\nabla\vec{v})_x : T_xM \to T_xM,$$
$$\vec{x} \mapsto \nabla_{\vec{x}}\vec{v} = (\nabla\vec{v})_x(\vec{x}).$$

The vector $\nabla_{\vec{x}}\vec{v}$ is the covariant derivative of vector field $\vec{v}$ at position x in direction $\vec{x}$, which is the tensor mapping result $\nabla\vec{v}(\vec{x})$.

Following [3], we can refer $\nabla\vec{v}$ to a basis $\{\vec{e}_i \otimes \omega^j\}$, and write the components as $n \times n$ matrix, $\nabla_j v^i$, with row and column indices i, j, respectively (n is the dimension of M, i.e., for a curved sphere $n = 2$). However, only when the manifold M is flat and Cartesian or affine coordinates are used, $\nabla_j v^i = \partial_j v^i$. The difference to the general case is a metric connection on M [12], corresponding to a notion of parallel transport [13] of vectors. The connection can be given in components using **Christoffel symbols** Γ^i_{jk}. We get $\nabla\vec{v}$ in components as:

$$\nabla_j v^i := \partial_j v^i + \Gamma^i_{jk} v^k. \tag{2.14}$$

Definition 6. *For a manifold M, the Christoffel symbols Γ^i_{jk} determine the covariant derivatives of the basis vector fields $\{\vec{e}_i\}$ on the manifold, at each point $x \in M$, referred back to the same basis $\{\vec{e}_i\}$ at x, i.e., $\nabla_{\vec{e}_j}\vec{e}_k = \Gamma^i_{jk}\vec{e}_i$, and therefore*

$$\Gamma^i_{jk} = \omega^i(\nabla_{\vec{e}_j}\vec{e}_k). \tag{2.15}$$

In general, a connection is not uniquely defined but needs to be chosen. Following

Rautek et al. [3], we employ the **Levi-Civita connection**, which can be derived intrinsically from the metric of M.

2.7 Rigid-body Motion and Isometry

2.7.1 Rigid-body Motion

Rigid-body motions are transformations which guarantee that the distance between any two points remains constant. Rigid-body dynamics are studied in 3-dimensional space in physics and computer graphics. The set of all possible rigid body motions in 3-dimensional space is called the Special Euclidean Transformation Group SE(3), including all the possible rotations and translations in three-dimensional space.

When an observer is undergoing rigid-body motion, the reference frame transformation will preserve the distance. Our framework enforces all candidate observers to be physically-realizable, which demands that the observers do time-dependent rigid-body motions.

2.7.2 Isometry

A map $\varphi : X \to Y$ on a manifold that does not change the distances of points is called **Isometry**. In our framework, we restrict observers to be non-deforming to the input flow field, which implies that the used reference frame transformations must be isometries.

2.8 Lie Group and Lie algebra

2.8.1 Lie Group

A group is defined [14] as:

Definition 7. *A group G is a set equipped with an operation $\cdot$ and an identity element e, satisfying the following properties:*

- $g_1 \cdot g_2 \in G$*, for all* $g_1, g_2 \in G$*.*

- $(g_1 \cdot g_2) \cdot g_3 = g_1 \cdot (g_2 \cdot g_3)$*, for all* $g_1, g_2, g_3 \in G$*.*

- $g \cdot e = e \cdot g$*, for all* $g \in G$*.*

- $g \cdot g^{-1} = g^{-1} \cdot g = e$*, for all* $g \in G$*.*

Definition 8. *A Lie group* G *is a group that is also a smooth manifold.*

Definition 9. *A Matrix Lie group* G *is a Lie group of* $n \times n$ *matrices (for some fixed* n*) that is closed under products, inverses, and non-singular limits [14].*

The Special Euclidean Transformation Group (SE(3)) is a Matrix Lie group (6-dimensional manifold). The Special Orthogonal Group (SO(3)), which we will use for observers on the sphere manifold, is another example of a Matrix Lie group. They can be treated as smooth manifolds, where every element corresponds to a 3-dimensional rotation or rigid-body motion matrix.

2.8.2 Lie Algebra

The Lie algebra is a vector space V over some number field F, with an additional vector product, the so-called Liebracket [,]. The Lie algebra satisfies the following properties:

- Closure: $\forall X, Y \in V, [X, Y] \in V$.

- Bilinearity: $\forall X, Y, Z \in V, \forall a, b \in F, [aX + bY, Z] = a[X, Z] + b[Y, Z]$.

- Alternativity: $\forall X \in V, [X, X] = 0$.

- Jacobi identity: $\forall X, Y, Z \in V, [X, [Y, Z]] + [Z, [X, Y]] + [Y, [Z, X]] = 0$.

- Anticommutativity: $\forall X, Y \in V, [X, Y] = -[Y, X]$.

The vector space structure means that a Lie algebra has a basis, i.e., a set of n linearly-independent basis vectors (basis vector fields) for an n-dimensional Lie algebra. The Lie algebra of rigid motions on the manifolds $M = \mathbb{R}^2$ and $M = \mathbb{S}^2$, respectively, are each a three-dimensional vector space. This implies that any element of the Lie algebra, i.e., any infinitesimal rigid motion, can be referred to three basis vectors, giving three corresponding real coefficients:

$$\vec{w}(x,t) = a(t)\,\vec{e}_1(x) + b(t)\,\vec{e}_2(x) + c(t)\,\vec{e}_3(x). \tag{2.16}$$

This expression is a linear combination of three basis vector fields with three real coefficients (a, b, c). For a time-dependent motion the coefficients are also time-dependent. Based on this, we model observers as the Lie algebra of Killing fields (See section 4.2).

In fact, the Lie algebra vector space is the tangent space of the smooth Lie group manifold at the identity element. More about the Lie algebra like the definitions of scalar multiplication, vector addition, linear independence, and the Lie bracket are give in Appendix A.

2.8.3 Matrix Exponential Map

Every Lie group is associated with a Lie algebra. For instance, matrix Lie group SO(3) is a smooth manifold, with the corresponding Lie algebra $\mathfrak{so}(3)$ which is the tangent space of the Lie group with a bi-linear, anti-symmetric bracket that satisfies the Jacobi identity. The Lie algebra will describe the local properties of the Lie group, the exponential map maps a Lie algebra $\mathfrak{g}$ to the corresponding Lie group G, i.e.,

$$\begin{aligned} exp &: \mathfrak{g} \to G, \\ X &\mapsto exp(X). \end{aligned} \tag{2.17}$$

For a matrix $\mathbf{M}$, the matrix exponential map reads:

$$exp(\mathbf{M}) = e^{\mathbf{M}} := \sum_{k=0}^{\infty} \frac{\mathbf{M}^k}{k!} = \mathbf{I} + \mathbf{M} + \frac{\mathbf{M}^2}{2!} + \frac{\mathbf{M}^3}{3!} + ..., \tag{2.18}$$

For example, we will use the exponential map of $\mathfrak{so}(3)$ to compute the observers' rigid motions SO(3).

2.9 Reference Frame Invariances

When we are extracting flow features (e.g., a vortex) from a flow field, we need to investigate the velocities relative to different reference frames (i.e., observers). An observer can be thought of as a camera that records velocities relative to its own motion. For many simulated flow fields, we cannot find one reference frame that follows all the vortices. However, we require extraction methods (like vortex detection methods) must be invariant under reference frame transformations.

2.9.1 Galilean Invariance

Galilean Invariance means a measure is invariant under constant-velocity translations of the reference frame and can be characterized by the following equation:

$$x^* = x + c_0 + tc_1,\ t^* = t - a, \tag{2.19}$$

where the point $p = (x, t)$ in space and time domain $[D \times T]$ is transformed to a new coordinate $p^* = (x^*, t^*)$ in the same space and time domain $[D \times T]$.

Most importantly, from equation 2.19, we can get an insight: quantities that compute from derivatives (spatial derivatives and temporal derivatives) are Galilean invariant, e.g., the acceleration of a vector field.

2.9.2 Objectivity

The second class of reference frame invariance is objectivity [8, 6, 7, 3]. A measure is objective if it remains unchanged when the reference frame undergoes rigid-body motions, including both translations and rotations. Such a transformation can be described by:

$$x^* = Q(t)x + c(t),\ t^* = t - a. \tag{2.20}$$

A formal definition of objectivity [15] reads:

Definition 10. *A scalar s is objective if it remains unchanged under any rigid-body reference frame transformation. A vector $\vec{v}$ is objective if Equation. 2.20 transforms it to $\vec{v}^* = Q(t)\vec{v}$. A second-order tensor T is objective if Equation. 2.20 transforms it to $T^* = Q(t)TQ(t)^T$.*

The objectivity of a measure is a desired property of a feature extraction techniques, as for most features the extraction methods rely on observers that perform both translation and rotation.

Chapter 3

Related Work

3.1 Flow Visualization

Flow visualization is one of the most crucial topics of scientific visualization, and many techniques have been developed over the years to visualize flow fields. Generally speaking, flow visualization techniques can be categorized into four types: direct visualization, texture-based visualization, geometric visualization and feature-based visualization.

Direct flow visualization does not carry out any major pre-processing of the input data but directly output the flow data. For example, vector glyph visualization shows the velocity of the vector field using arrows or cones and gives a straightforward visualization [16].

Texture-based flow visualization represents flow data as textures and tries to convey characteristics of the flow, such as its directional motion with pixel data. For instance, line integral convolution (LIC) [10] integrates a random noise texture along one streamline for each texel. It convolves the noise along streamlines using a filter kernel, the resulting pattern can intuitively show instantaneous direction of the flow field.

Geometric flow visualization methods use different geometric objects whose characteristics exhibit the underlying properties of the flow field. In our framework, we widely use stream objects like streamlines and path lines [17, 18] to indicate vortex structures.

Feature-based flow visualization introduces a higher level of abstraction where an extraction step precedes the visualization. Examples include shock waves, separation and attachment lines [19].

3.2 Reference Frames

Flow fields are velocity vector fields, and therefore they can only be given relative to a frame of reference, or an observer. This fact is important in the detection and visualization of vortices, where it has become a major consideration when evaluating different vortex detection methods [6, 8].

In order to visualize and detect vortices in unsteady flow, there is a lot of research that is aimed at finding a suitable observer in which the flow appears nearly steady. For example, Bhatia et al. [20] used the Helmholtz-Hodge decomposition (HHD) to determine a suitable reference frame. Wiebel et al. [21] proposed a method based on an invariant of HDD that decomposes the flow into localized and potential components. A new observer is extracted by subtracting the potential flow components from the input field. The resulting relative field is a nearly steady observer field.

Consistency of conclusions with respect to a particular kind of reference frame transformation is called invariance, including Galilean invariance [22], rotation invariance [23], or invariance relative to rigid motions. The latter is also known as objectivity [6] or frame indifference [15]. This concept is originally defined in continuum mechanics [15, 24, 25], and later is recognized to be of fundamental importance for vortex detection [6, 1, 7, 8], although used earlier as well [26, 27].

3.3 Multiple Reference Frames

Most commonly, only a single reference frame (observer) is considered for flow visualization or computational tasks. However, for many real-world flow phenomena, it is not sufficient to explore a fluid field relative to just one observer. For example, when there are vortices in a flow field moving in different directions. In this case, it's obvious that there is no global reference frame that can track their different motions and detect all features of interest. For this reason, Günther et al. [7] proposed to compute point-wise local reference frames that are suitable to track vortices for their local neighborhood. This method was later extended to affine transformations [28]. Recent research proposed to average a finite set of observers origin from the Galilean-invariant critical points [29], one observer per point in space-time [7], or a continuous field of observers [2, 3].

Novel approaches have integrated techniques that are not commonly used in flow visualization to find a suitable observer, e.g., Kim and Günther [30] used deep learning techniques to extract robust reference frames for vortex extraction under the presence of noise. A convolutional neural network (CNN) is used to extract local reference frames based on prior training knowledge. In these local reference frames, vortices are no longer hidden by ambient flow.

Hadwiger et al. [2] compute a field of observers using a global optimization method that minimizes the observed time derivative as well as the Killing energy (a measure for how much the field is deforming the input field). The resulting approximate Killing field is used to transform a given input flow.

3.4 Vortex Detection

3.4.1 Vortex Definition

For many research and engineering areas, vortices tend to be the most prominent features of fluid flows, as they can have various impacts on fluid dynamics. However, even though there is a large number of studies about vortices in the past, the definition of vortex remains unsettled until today. There is no universal and fixed agreement on what constitutes a vortex structure, and a range of definitions have been proposed. The first definition about vortex given by Lugt [31] is the following:

Definition 11. *A vortex is the rotating motion of a multitude of material particles around a common center.*

A more concrete definition proposed by Robinson [32] is:

Definition 12. *When viewed from a reference frame co-moving with the center of the vortex core, a vortex exists when instantaneous streamlines that are mapped onto a plane normal to the vortex core, exhibit a roughly circular or spiral pattern.*

These two definitions share a similar idea that a vortex can be viewed as a swirling motion over time that is centered around a time-dependent curve called vortex core line, which can be viewed clearly with a suitable reference frame. The vortex core line describes how the center of a vortex is moving as time goes. These concepts have been widely used in later research throughout the years and have led to numerous approaches for identifying vortices. In the following sections, we will present some existing numerical vortex detection methods relevant to this work. Generally speaking, vortex detection criteria can be grouped into line-based and region-based techniques.

3.4.2 Line-based Detection Methods

Most of the line-based techniques are based on an important insight about vortex structures: even though the region of a vortex lacks a clear-cut definition, vortex core lines can be concretely defined as the lines that describe how a vortex center is moving over time, i.e., the vortex core line is the trace of the vortex center.

In the steady flow field, line-based vortex detectors have been proposed in recent years. For instance, the predictor-corrector technique tries to predict where the vortex will move, based on pressure minimum. Core lines are integrated with correction for wrong-predictions [33]. Sujudi and Haimes [34] have proposed a popular reduced velocity criterion: $\vec{v} - (\vec{v}^T e)e = 0$. This formulation will locate points where the velocity gradient tensor has one real and two complex eigenvalues and the flow vector is in the direction of the eigenvector that corresponds to the real eigenvalue. Another approach called the parallel vectors (PV) operator, proposed by Peikert and Roth [35], is based on the work of Sujudi and Haimes [34]. This approach derives two vector fields from the three dimensional flow field and afterwards, the vortex core lines are located at the points where the two vector fields are parallel. If we describe the approach of Sujudi-Haimes using the PV operator, it can be rephrased as $\vec{v} \parallel \nabla\vec{v}(\vec{v})$.

In the time-dependent flow field, early approaches like Bauer et al. [36] and Theisel et al. [37] tend to track the core lines of vortex structures, other methods were based on the method of Sujudi-Haimes. For example, Fuchs et al. [38] and Weinkauf et al. [22] extended the method of Sujudi-Haimes and the method of PV operator to the unsteady field by characterizing swirling motion in 2D and 3D flows with a novel mathematical characterization. Fuchs conducted a comprehensive study about flow field features and provided interactive visual analysis of flow field [38] by extending and integrating vortex detectors into the visual analysis process.

3.4.3 Region-based Detection Methods

Region-based detection techniques are methods that generally compute scalar quantities like vorticity, deviation, pressure, to detect the region of a vortex. Most region-based detection techniques give straightforward results by setting suitable thresholds of those scalar quantities.

One obvious measure to detect a vortex region is high vorticity magnitude, since it denotes the tendency of local rotations in the fluid flow. A vortex region always has a high vorticity magnitude, but the converse does not hold. Vorticity does not discern between pure shearing motions and swirling motions [39, 40]. Low pressure is used as another indicator of a vortex. However, this only holds in steady inviscid 2D flows [41]. Kasten et al. [42] proposed to use vanishing acceleration to identify vortex regions, and this criterion has an advantage that such a measure does not require any threshold.

Most commonly used region-based methods are based on the analysis of the velocity gradient tensor and its decomposed parts. For example, the Q-criterion as proposed by Hunt et al. [43] identifies vortices as regions where the second invariant Q of the velocity gradient tensor is positive. The λ_2 criterion of Jeong and Hussein [44] identifies vortices based on the eigenvalues of the symmetric matrix $\mathbf{S}^2 + \mathbf{\Omega}^2$. These methods only use spatial (in contrast to time) derivatives, which can lead to challenges for unsteady flow. One reason for this is that many early methods are partially based on critical point theory of steady vector fields [45, 46].

More recently, the objectivity of vortex criteria is seen to be a crucial and desired property of vortex detectors. Haller et al. [1] defined vortices objectively using the original vorticity tensor $\mathbf{\Omega}$ and introduced the instantaneous vorticity deviation (IVD). Integrating this measure along a path line produces the Lagrangian-averaged vorticity deviation (LAVD). Because the deviation is objective relative to a specific global observer, the resulting IVD and LAVD are objective criteria.

In our experiments, λ_2 [44] criterion and Lagrangian-Averaged Vorticity Deviation (LAVD) [1] are used and more detail about these two methods will be introduced in chapter 6.

3.5 Generic Objectivization

As vortex extraction is based on reference frames, one of the most desirable properties of a vortex detector is called objectivity, which means: although the relative velocity fields are different, the conclusions about vortex criteria are independent of choice of the observer.

The work of Günther et al. [7] connected the task of developing vortex detection methods that are objective with solving an optimization problem that enables **generically "objectivizing"** non-objective vortex criteria. Günther et al. [47] proposed a concept called hyper-objectivity, a more strict condition than objectivity: the invariance under any rigid-body motions, uniform and smooth scaling. Based on the idea of solving an optimization problem to determine objective observers, Hadwiger et al. [2] carried out a global optimization procedure to extract observers from Killing fields. Rautek et al. [3] extended the optimization and the use of Killing fields to represent observers in curved space.

However, the objectivity of these approaches, by the definition of objectivity, only guarantees that any physically-realizable, i.e., rigidly-moving, observer will reach the same conclusions regarding the "joint" detection of vortices. None of these methods guarantees that there is **one physically-realizable observer** that by itself (i.e., not jointly) would have computed the same vortex using physically-motivated criteria. While this does not change the fact that the above methods are provably objective [7, 2, 3, 48], recent work has argued that physical criteria should also be used to determine the validity and consistency of vortex detection [49].

In this thesis, we tackle this problem from a different perspective and combine the advantages of both viewpoints: We enable the interactive exploration of multiple, objectively-determined observers, but allow users to choose specific, physically-realizable observers for actual vortex detection and visualization. In this way, physically-motivated criteria fully apply.

3.6 Killing Vector Fields

A Killing vector field (often called Killing field), named after Wilhelm Killing, is a vector field on a Riemannian manifold (or pseudo-Riemannian manifold) that preserves the metric. It corresponds to infinitesimal isometries on a Riemannian manifold. The properties of Killing fields are extensively explored in Riemannian geometry [50, 13, 51]. They have been used explicitly in geometry [52, 53, 54, 55], including vector field design on curved surfaces [56, 57]. Recently the concept of Killing fields was introduced to the flow visualization community [2, 3].

Chapter 4

Observers and Sets of Observers

To guarantee that the results we extract from a flow field are meaningful, our approach is based on a set of **physically-realizable** observers. In this chapter, we will describe what physically-realizable observers are, how these observers are modeled and how the corresponding description can be extracted from an arbitrary observer velocity field $\vec{u}$.

4.1 Physically-Realizable Observers

Definition 13. *An observer is physically realizable if its time-dependent motion relative to any other physically-realizable observer is a rigid motion.*

This basically means that every rigid reference frame motion corresponds to a physically-realizable observer. The reason for the "circular" argument above is that all motion is relative, and thus for a rigid motion to describe a physical observer, the background reference frame (e.g., the lab frame) must already be known to be physically realizable. In terms of our input field $\vec{v}$, this means that the whole field must have been measured relative to the same, physically-realizable observer.

A physically-realizable observer is an observer that undergoes a time dependent rigid-body motion relative to other physically-realizable observers including the lab frame observer.

Reference frame transformation between two observers is one-to-one mapping and can be mathematically defined as a diffeomorphism. Because our framework guarantees that all candidate observers are physically realizable, we only allow transformations that preserve distances, which implies that the diffeomorphism is in fact an isometry.

4.2 Killing Vector Fields

4.2.1 Killing Fields and Isometry

A Killing vector field [51, 50] is a vector field on a Riemannian manifold that corresponds to the **infinitesimal isometries** of a given manifold M.

Definition 14. *A vector field $x \mapsto \vec{w}(x)$ is a Killing field, if its spatial velocity gradient $\nabla\vec{w}$ is identically anti-symmetric. That is, the inner product:*

$$\langle \nabla\vec{w}(\vec{x}), \vec{x} \rangle = 0. \tag{4.1}$$

This must hold at all points $x \in M$, for all vectors $\vec{x}$. Eq. 4.1 is valid for all general manifolds M, with $\nabla\vec{w}$ the covariant derivative of $\vec{w}$ (in components, $\nabla_j w^i$; see section 2.6.4 for detail). For Euclidean space $M = \mathbb{R}^2$ and Cartesian coordinates, $\nabla\vec{w}$ is simply a Jacobian matrix $\partial_j w^i$ of partial derivatives. For $M = \mathbb{S}^2$, we employ the intrinsical **Levi-Civita connection**, which gives the **Christoffel symbols** on this manifold.

Killing vector fields correspond to the derivatives of rigid body motions on a manifold and therefore do not distort distances.

4.2.2 The Lie Algebra of Killing Fields

In order to manipulate observers and guarantee physically realizability at the same time, our mathematical framework models any possible physically-realizable observer motion as an element of the Lie algebra (in terms of the vector space structure, a "vector") of Killing fields on our manifolds of interest, the plane $\mathbb{R}^2$ and the sphere $\mathbb{S}^2$.

It is worth noting that the Lie algebra is a vector space of vector fields, i.e., each element (abstract vector) of this Lie algebra vector space is itself a whole **vector field** on the underlying manifold M. This Lie algebra, therefore, comprises all possible observer motions. Interestingly, as we will see in sections 4.2.3 and 4.2.4, the Lie algebra on the plane $\mathbb{R}^2$ and the sphere $\mathbb{S}^2$, both are three-dimensional vector spaces.

Dimensionality and Basis Expansion

Each Killing field can be referred to a basis comprised of **basis vector fields**. It is a linear combination of basis vector fields.

Theorem 1. *A Killing field $\vec{w}$ on a manifold M is uniquely defined by a spanning set of* ***basis Killing fields*** *$\{\vec{e}_i\}$ and the corresponding set of coefficients $\{w^i\}$. The Killing field $\vec{w}$ is then given as $\vec{w} = \sum_i w^i \, \vec{e}_i$.*

In this thesis, we use vector spaces of Killing fields that are three-dimensional, and therefore we only need three coefficients w^i, which we will instead denote by $a := w^1, b := w^2, c := w^3$, or collectively as (a, b, c).

Inner Products of Vector Fields

The orthogonality of basis functions should be verified and we also need to compare candidate observers by comparing their corresponding Killing vector fields, all of these tasks depend on a definition of the inner product between two vector fields. Thus, we define the inner product between two (Killing) vector fields $\vec{w}_1$ and $\vec{w}_2$, with domain D, as the integration of inner products:

$$\langle\langle \vec{w}_1, \vec{w}_2 \rangle\rangle := \int_D \langle \vec{w}_1, \vec{w}_2 \rangle_x \; dA. \tag{4.2}$$

Here, the usual inner product in the tangent space at each point $x \in D$ is denoted by $\langle \cdot, \cdot \rangle_x$, and dA denotes integration over area elements. This inner product is

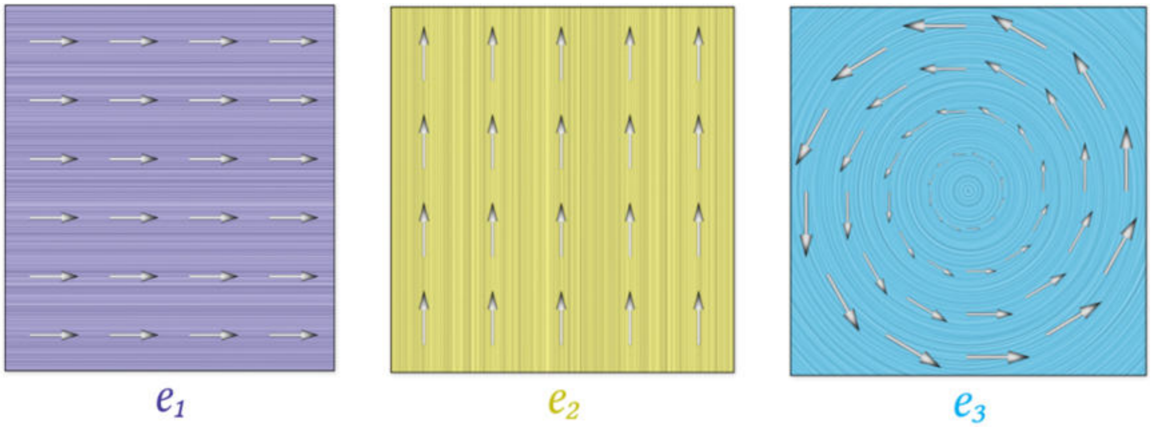


Figure 4.1: **Physically-realizable observer motions referred to basis Killing fields on the plane.** The Lie algebra of physically-realizable observer motions on the plane is a three-dimensional vector space, where each element $\vec{w}$ can be referred to a basis of three linearly-independent basis Killing vector fields $\{\vec{e}_1, \vec{e}_2, \vec{e}_3\}$, such as those shown here.

defined with respect to a domain $D \subseteq M$. For $M = \mathbb{S}^2$, we use $D = \mathbb{S}^2$, because $\mathbb{S}^2$ is compact and the integral is therefore guaranteed to be finite. For $M = \mathbb{R}^2$, we typically integrate over a compact rectangular subdomain $D = [x_a, x_b] \times [y_a, y_b] \subset \mathbb{R}^2$.

4.2.3 Basis Killing Fields in the Euclidean Plane

The Lie algebra of rigid-body motion on the euclidean plane $M = \mathbb{R}^2$ is a three-dimensional vector space. There are infinitely many basis vector fields on the Euclidean plane to compose Killing fields. We determine the following three linearly independent basis vector fields in $\mathbb{R}^2$, where the vectors at any point $x = (\hat{x}, \hat{y}) \in \mathbb{R}^2$ are:

$$\vec{e_1}(\hat{x},\hat{y}) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \vec{e_2}(\hat{x},\hat{y}) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \vec{e_3}(\hat{x},\hat{y}) = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix}, \tag{4.3}$$

Fig. 4.1 depicts the three basis Killing fields $\{\vec{e}_1, \vec{e}_2, \vec{e}_3\}$ on the Euclidean plane given by Eq. 4.3. Using this basis, we can therefore write any Killing field $\vec{w}$ on $M = \mathbb{R}^2$ as $\vec{w} = a\,\vec{e}_1 + b\,\vec{e}_2 + c\,\vec{e}_3$, with three coefficients (a, b, c). We note that, due to this particular choice of basis, (a, b) have the meaning of a linear velocity vector (velocity for translation), and the third coefficient c has the meaning of angular velocity (velocity for rotation). In fact, the linear velocity is the same constant vector at all points $x \in \mathbb{R}^2$, the angular velocity basis vectors, on the other hand, are different from point to point.

Orthogonality. Due to symmetry, this basis is orthogonal for a rectangular domain of integration with $(\hat{x}_0, \hat{y}_0)$ chosen in the center, where, in particular, we have $\langle\langle \vec{e}_1, \vec{e}_3 \rangle\rangle = \langle\langle \vec{e}_2, \vec{e}_3 \rangle\rangle = 0$, because $\langle \vec{e}_1(\bar{x}, \bar{y}), \vec{e}_3(\bar{x}, \bar{y}) \rangle = -\langle \vec{e}_1(-\bar{x}, -\bar{y}), \vec{e}_3(-\bar{x}, -\bar{y}) \rangle$, and likewise for the fields $\vec{e}_2$ and $\vec{e}_3$. The orthogonality $\langle\langle \vec{e}_1, \vec{e}_2 \rangle\rangle = 0$ immediately follows from the orthogonality $\langle \vec{e}_1(\hat{x}, \hat{y}), \vec{e}_2(\hat{x}, \hat{y}) \rangle = 0$ at every point $(\hat{x}, \hat{y})$. See Eq. 4.2 for the definition of the vector field inner product $\langle\langle \cdot, \cdot \rangle\rangle$.

4.2.4 Basis Killing Fields on the Sphere

The Lie algebra of Killing fields on $M = \mathbb{S}^2$ is also three-dimensional. We define the following three basis Killing fields for the two-sphere $\mathbb{S}^2 := \{(\hat{x}, \hat{y}, \hat{z}) | \hat{x}^2 + \hat{y}^2 + \hat{z}^2 = 1\}$ embedded in $\mathbb{R}^3$, where the vectors at any point $x = (\hat{x}, \hat{y}, \hat{z})$, as elements of the tangent space embedded in $\mathbb{R}^3$ at that point, are given by (see Fig. 4.2)

$$\begin{aligned}
\vec{e}_1(\hat{x}, \hat{y}, \hat{z}) &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix}, \\
\vec{e}_2(\hat{x}, \hat{y}, \hat{z}) &= \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix}, \\
\vec{e}_3(\hat{x}, \hat{y}, \hat{z}) &= \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix}.
\end{aligned} \tag{4.4}$$

Fig. 4.2 depicts the three basis Killing fields $\{\vec{e}_1, \vec{e}_2, \vec{e}_3\}$ on the sphere given by

Eq. 4.4, and any Killing field $\vec{w}$ on $M = \mathbb{S}^2$ can be written as the linear combination with coefficients (a, b, c) referred to this basis, i.e., $\vec{w} = a\,\vec{e}_1 + b\,\vec{e}_2 + c\,\vec{e}_3$. Below the spheres, the corresponding anti-symmetric matrices are shown.

We note that, due to the above choice of basis, the coefficients (a, b, c) determine a 3D angular velocity vector, and $\omega^2 := a^2 + b^2 + c^2$ is the corresponding (squared) angular velocity (magnitude). Since Eq. 4.1 is intrinsically defined in each tangent space, to see that the 2D tensors $\nabla\vec{e}_i$ are anti-symmetric, we must compute the covariant derivatives $\nabla\vec{e}_i$ at all $x \in S^2$. Using a right-handed orthonormal basis in each tangent plane, they are

$$(\nabla\vec{e}_i)_x = \begin{bmatrix} 0 & -\cos\varphi_i(x) \\ \cos\varphi_i(x) & 0 \end{bmatrix}. \tag{4.5}$$

Here, the angle $\varphi_i(x) \in [0, \pi]$ is the colatitude of x away from the "north pole" of the respective $\vec{e}_i$, i.e., $\hat{x} = 1$ for $\vec{e}_1$, $\hat{y} = 1$ for $\vec{e}_2$, $\hat{z} = 1$ for $\vec{e}_3$. Therefore, Eq. 4.1 again holds for all $\vec{e}_i$, i.e., $\langle\nabla\vec{e}_i(\vec{x}), \vec{x}\rangle = 0$ for all $\vec{x}$.

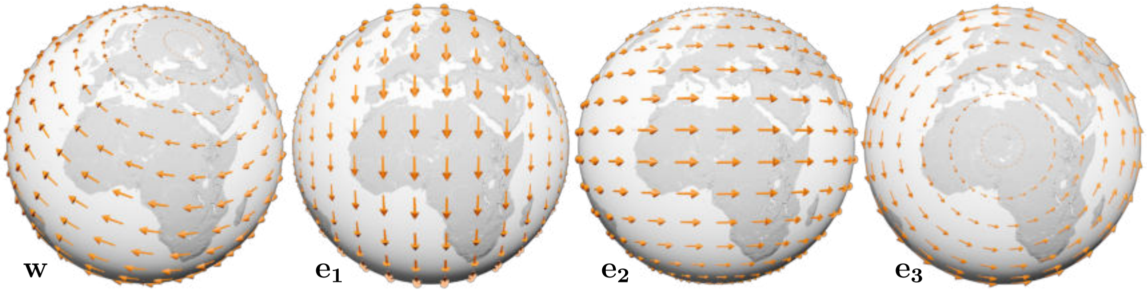
Orthogonality. This basis is also orthogonal, i.e., $\langle\langle\vec{e}_i, \vec{e}_j\rangle\rangle = 0$ for all $i \neq j$. This can be confirmed by analytic integration over the sphere.

4.2.5 Time-dependent Killing Fields

Just like we have steady vector fields and unsteady vector fields, if a Killing field is varying with time, then it is a **time-dependent** Killing field with the properties talked above to hold for each fixed time t, giving a time-dependent field:

$$(x, t) \mapsto \vec{w}(x, t). \tag{4.6}$$

Theorem 2. *A time-dependent Killing field $\vec{w}(x, t)$ gives the derivative of a time-dependent rigid motion. Therefore, it corresponds to a physically-realizable observer,*



$$W = \begin{bmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{bmatrix} \quad E_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad E_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \quad E_3 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Figure 4.2: **Physically-realizable observer motions referred to basis Killing fields on the sphere.** The Lie algebra of physically-realizable observer motions on the sphere is a three-dimensional vector space, where each element $\vec{w}$ can be referred to a basis of three linearly-independent basis Killing vector fields $\{\vec{e}_1, \vec{e}_2, \vec{e}_3\}$, such as those shown here.

defined by integrating the time-dependent Killing field.

As above, it is important that $\vec{w}$ is measured relative to a physically-realizable observer. We will also exploit the following important theorem:

Theorem 3. *A Killing field $\vec{w}$ is uniquely defined by: (1) a single vector $\vec{w}(x)$ at an arbitrarily chosen point $x \in M$, and (2) the corresponding (anti-symmetric) velocity gradient $(\nabla \vec{w})_x$ at the same point x.*

This is a well-known property of Killing fields [50, p.315]. However, this property is crucial to objectively "extracting" physically-realizable observers from the methods described in prior work [7, 2, 3], and converting them to the representation of our framework.

4.3 Representation of Observers

We define an observer as the time-dependent, rigid reference frame motion given by a time-dependent Killing field $(x, t) \mapsto \vec{w}(x, t)$. The time-dependent Killing fields give us **infinitesimal isometries** on the given manifold, the integration (matrix exponential map) of the time-dependent Killing fields defines a time-dependent rigid motion on the manifold.

Instead of storing an explicit vector field $(x, t) \mapsto \vec{w}(x, t)$ for each observer, we store observers as the time dependent components $t \mapsto (a(t), b(t), c(t))$ with respect to a chosen basis of Killing fields. While in principle this basis is arbitrary, we use the basis vector fields given by Eq. 4.3 (for Euclidean domains $M = \mathbb{R}^2$), and Eq. 4.4 (for spherical domains $M = S^2$), respectively. Thus, all possible observer motions are fully determined by coefficients referred to a three-dimensional basis.

We fix a basis of three **basis Killing vector fields** $\{\vec{e}_1, \vec{e}_2, \vec{e}_3\}$ on M in advance, then represent every possible observer motion by three real coefficients (a, b, c) for each time t. Any observer is therefore determined by a function

$$t \mapsto \big(a(t), b(t), c(t)\big). \tag{4.7}$$

The corresponding observer motion is given by a time-dependent Killing vector field $\vec{w}$, determining the motion via its derivative, as

$$(x, t) \mapsto \vec{w}(x, t) = a(t)\,\vec{e}_1(x) + b(t)\,\vec{e}_2(x) + c(t)\,\vec{e}_3(x). \tag{4.8}$$

We make heavy use of the vector space properties and define operations between time dependent Killing Fields such as interpolation, averaging, scaling, simply by performing these operations on the coefficients (a, b, c) that represent the observers.

Finally, any given observer $\vec{w}$ is thus given by a function $t \mapsto ((a(t), b(t), c(t))$ (Eq. 4.7), which at any time allows us to obtain the corresponding time-dependent Killing field $\vec{w}(x, t)$ from Eq. 4.8.

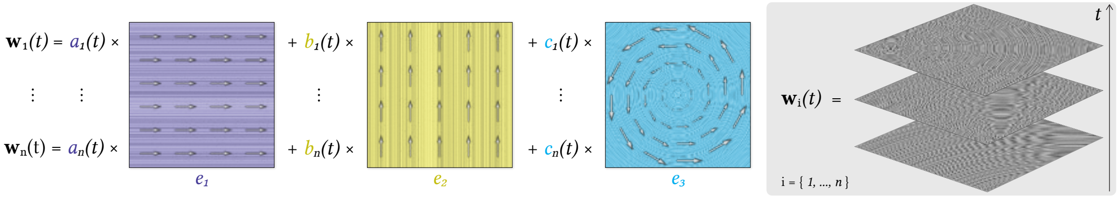


Figure 4.3: **Observer representation.** Any physically-realizable observer is determined by a time-dependent Killing field $(x, t) \mapsto \vec{w}(x, t) = a(t)\,\vec{e}_1(x) + b(t)\,\vec{e}_2(x) + c(t)\,\vec{e}_3(x)$. The basis vector fields $\vec{e}_1$ (purple), $\vec{e}_2$ (yellow), and $\vec{e}_3$ (cyan) are steady vector fields that do not change over time. Any possible (time-dependent) observer $\vec{w}_i$ is solely determined by a time-dependent function $t \mapsto (a_i(t), b_i(t), c_i(t))$ of three scalar coefficients (a_i, b_i, c_i) per time t.

4.3.1 Comparing Observers

We exploit the low-dimensional (in our case, three-dimensional) vector space structure of Killing vector fields in order to efficiently quantify the similarity of two observers given as time-dependent Killing fields.

First, from the definition of the inner product of two vector fields given by Eq. 4.2, we can define the difference between two arbitrary (time-independent) Killing fields as the scalar-valued distance function

$$d(\vec{w}_1, \vec{w}_2) := \sqrt{\langle\langle \vec{w}_1 - \vec{w}_2, \vec{w}_1 - \vec{w}_2 \rangle\rangle}. \tag{4.9}$$

This definition in fact defines a **metric** on the vector space of all Killing fields in the domain $D \subseteq M$, where we want to compare observers. For time-dependent Killing fields $\vec{w}_1(x, t)$ and $\vec{w}_2(x, t)$, we now define

$$d(\vec{w}_1(t), \vec{w}_2(t); t_0, t_1) := \int_{t_0}^{t_1} d\big(\vec{w}_1(t), \vec{w}_2(t)\big)\, dt. \tag{4.10}$$

Here, we have integrated over a desired time window $[t_0, t_1]$, over which we want to compare the two observers. The rationale for this definition is simply that we integrate the per-time distance values over a time window of interest, in order to quantify the

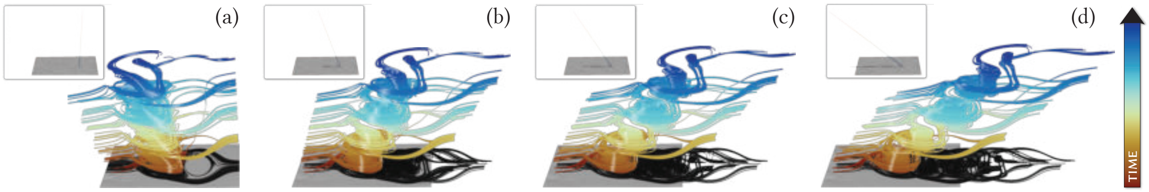


Figure 4.4: **Cylinder flow.** The vortex structures in this data set are short-lived. They appear and disappear frequently over time. In the input lab frame (a) different vortices overlap (over time) around the same spatial position and appear to be the same vortex. We interpolate (a-d) between the lab frame and an objectively-determined constant-velocity observer that "co-moves" with the vortices, revealing them more clearly. The insets depict observer world lines. Time corresponds to the vertical axis and is also color-coded to help highlight how individual vortices appear and disappear over time.

"total" difference between the two observers over that time window. (Longer time windows, in general, give larger distances, when the two fields are different.)

It is worth noting that both $\vec{w}_1$ and $\vec{w}_2$ are objectively-determined observers in our framework, then the difference equations above give **objective** scalars.

4.3.2 Determination of Observer Sets

We can store a set of observers by storing multiple observers, each represented by a function $t \mapsto \big(a(t), b(t), c(t)\big)$, as above (Eq. 4.7).

However, we can also "extract" an arbitrary number of observers from a given vector field, which we will refer to as the **observer field** $\vec{u}(x,t)$.

Observer World Lines

Theorem 3 shows that an observer can also be specified by a function

$$t \mapsto \Big(x(t), \vec{w}\big(x(t), t\big), \big(\nabla \vec{w}\big)_{(x(t),t)}\Big), \tag{4.11}$$

where $t \mapsto x(t)$ is the world line of the observer (the trajectory of the observer) on the manifold M, and we prescribe the corresponding vectors $\vec{w}$, and tensors $\nabla \vec{w}$

along this path. Although compared to Eq. 4.7, the above representation requires more storage, and will not be used in implementation, but this way of specifying Killing fields enables the general approach for **extracting rigid observers from a given (possibly non-rigid) observer velocity field** $\vec{u}$, which will be explained in section 4.3.3 and 4.3.4.

Observer World Lines from Observer Fields

We do not explicitly store the trajectory of the observer $t \mapsto x(t)$, as we can define the trajectory line of the observer as the path line of a particle released in the observer field $\vec{u}(x(t), t)$, which is obtained by solving the non-autonomous ODE:

$$\frac{d}{dt}x(t) = \vec{u}\big(x(t), t\big), \tag{4.12}$$

with an initial value $x(t_0)$, i.e., some chosen position (the seed position of the observer determined by the users) on the path line $t \mapsto x(t)$ at time $t = t_0$, for some chosen time t_0. We call this path $t \mapsto x(t)$ a **world line** [2] of the corresponding observer. In order to extract an observer, at each time t we refer to the corresponding point $x(t)$.

4.3.3 Observer Extraction in the Euclidean Plane

Given a single vector $\vec{u}(x)$ and the velocity gradient $\big(\nabla\vec{u}\big)_x$ at any point $x \in M = \mathbb{R}^2$, we can extract the coefficients of the corresponding Killing field $\vec{w}$, with respect to the basis fields $\{\vec{e}_1, \vec{e}_2, \vec{e}_3\}$ (Eq. 4.3), as

$$\nabla\vec{w} = \frac{1}{2}\big((\nabla\vec{u})_x - (\nabla\vec{u})_x^T\big) = \begin{bmatrix} 0 & -c \\ c & 0 \end{bmatrix}, \quad \vec{u}(x) + \nabla\vec{w}(\vec{r}) = \begin{bmatrix} a \\ b \end{bmatrix}. \tag{4.13}$$

We define $\vec{r} := (\hat{x}_0 - \hat{x}, \hat{y}_0 - \hat{y})^T$, with $(\hat{x}, \hat{y})$ the Cartesian coordinates of the point x, and $(\hat{x}_0, \hat{y}_0)$ the center point used in the definition of the basis vector field $\vec{e}_3$. The coefficients (a, b, c) now determine the Killing field $\vec{w} = a\,\vec{e}_1 + b\,\vec{e}_2 + c\,\vec{e}_3$, referred to the basis given in Eq. 4.3.

4.3.4 Observer Extraction on the Sphere

Given a single vector $\vec{u}(x)$ and the covariant derivative $\left(\nabla\vec{u}\right)_x$ at any point $x \in M = S^2$, we can extract the coefficients of the corresponding Killing field $\vec{w}$, with respect to the basis fields $\{\vec{e}_1, \vec{e}_2, \vec{e}_3\}$ (Eq. 4.4), using the following approach. Referring to the point $x \in S^2$ as the position vector $\vec{r}(x) := (\hat{x}, \hat{y}, \hat{z})^T$ in Cartesian coordinates, we can define the following orthonormal basis, written as column vectors of the matrix

$$\hat{\mathbf{B}} := \left[\; \vec{x} \;\middle|\; \vec{z} \times \vec{x} \;\middle|\; \vec{z} \;\right], \quad \vec{x} := \vec{u}(x)/\|\vec{u}(x)\|, \quad \vec{z} := \vec{r}(x). \tag{4.14}$$

When $\|\vec{u}(x)\| = 0$, $\vec{x}$ can be any unit vector in the tangent plane at x. Given the covariant derivative $\nabla\vec{u}$ in components as $\nabla_j u^i$ in some coordinate chart with tangent space basis $\{\vec{b}_1, \vec{b}_2\}$ at x, and the coordinate Jacobian $\mathbf{J}$ transforming from that basis to the basis $\hat{\mathbf{B}}$, we can compute[1]

$$\nabla_j \hat{u}^i = \mathbf{J}\left(\nabla_j u^i\right)\mathbf{J}^{-1}, \quad \nabla_j \hat{w}^i = \frac{1}{2}\left(\nabla_j \hat{u}^i - \nabla_i \hat{u}^j\right) = \begin{bmatrix} 0 & -\hat{c} \\ \hat{c} & 0 \end{bmatrix}. \tag{4.15}$$

Given the chart tangent space basis $\{\vec{b}_1, \vec{b}_2\}$, embedded in $\mathbb{R}^3$, the Jacobian $\mathbf{J}$ is given by the top-left 2×2 submatrix of the 3×3 matrix

$$\hat{\mathbf{J}} = \hat{\mathbf{B}}^T \left[\; \vec{b}_1 \;\middle|\; \vec{b}_2 \;\middle|\; \vec{z} \;\right]. \tag{4.16}$$

[1] We note that the anti-symmetrization used here depends on the fact that the basis $\hat{\mathbf{B}}$ is orthonormal, and hence the corresponding metric tensor is the identity. Otherwise, $\nabla_j u_i$ and the metric g_{ij} would have to be used explicitly. See [3].

From the obtained $\hat{c}$, and the length $\hat{b} := \|\vec{u}(x)\|$, we can now compute

$$\hat{\mathbf{K}} = \begin{bmatrix} 0 & -\hat{c} & \hat{b} \\ \hat{c} & 0 & 0 \\ -\hat{b} & 0 & 0 \end{bmatrix}, \quad \mathbf{K} = \hat{\mathbf{B}}\,\hat{\mathbf{K}}\,\hat{\mathbf{B}}^T = \begin{bmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{bmatrix}. \tag{4.17}$$

The coefficients (a, b, c) obtained from the matrix $\mathbf{K}$ now determine the Killing field $\vec{w} = a\,\vec{e}_1 + b\,\vec{e}_2 + c\,\vec{e}_3$, referred to the basis given in Eq. 4.4.

For completeness, we note that here $\hat{b} = \omega \sin\varphi$ and $\hat{c} = \omega\cos\varphi$, where $\omega^2 = \hat{b}^2 + \hat{c}^2 = a^2 + b^2 + c^2$ is the (squared) angular velocity of the global rotation of the sphere given by $\vec{w}$, with φ the colatitude of x relative to the critical point of $\vec{w}$ that is the "north pole" of the rotation, i.e., the point $(\bar{x}, \bar{y}, \bar{z})^T = (a, b, c)^T / \|(a, b, c)^T\|$, defined when $\omega \neq 0$.

4.4 Observer Averaging and Interpolation

Using the vector space structure of all observer Killing fields, we can directly average a set of observers and interpolate between two observers. When interpolate an arbitrary observer with the lab frame observer, the interpolation is identical to a scaling of the components $a(t), b(t), c(t)$. Most importantly, our mathematical framework guarantees that the resulting observers from interpolation or averaging are physically realizable.

4.4.1 Observer Averaging

Given a set of observers $\{\vec{w}_i\}_{i\in I}$ with the index $i \in I$ from some index set I, considering the vector space structure of the space of all observers, we can simply define the

average observer of this set of observers as

$$t \mapsto \big(\bar{a}(t), \bar{b}(t), \bar{c}(t)\big) := \frac{1}{|I|} \left(\sum_{i \in I} a_i(t), \sum_{i \in I} b_i(t), \sum_{i \in I} c_i(t) \right), \tag{4.18}$$

where $|I|$ is the cardinality of the index set I. If I is a whole region of a manifold M instead of a discrete set, we define the above average with integrals instead of sums, and the cardinality for normalization by an appropriate **measure** of the set I, e.g., the area measure on M.

Sec. 4.6.3 describes an application of this averaging technique, applied over a region $I = U(t) \subset \mathbb{R}^2$, with the standard area measure on $\mathbb{R}^2$.

4.4.2 Direct Observer Interpolation (Blending)

Given two observers $\vec{w}_1 = a_1\, \vec{e}_1 + b_1\, \vec{e}_2 + c_1\, \vec{e}_3$ and $\vec{w}_2 = a_2\, \vec{e}_1 + b_2\, \vec{e}_2 + c_2\, \vec{e}_3$, we can directly interpolate their coefficients via ($\alpha \in [0, 1]$)

$$\begin{aligned} \vec{w}(t;\alpha) = &\big((1-\alpha)\, a_1(t) + \alpha\, a_2(t)\big)\, \vec{e}_1 + \\ &\big((1-\alpha)\, b_1(t) + \alpha\, b_2(t)\big)\, \vec{e}_2 + \\ &\big((1-\alpha)\, c_1(t) + \alpha\, c_2(t)\big)\, \vec{e}_3. \end{aligned} \tag{4.19}$$

When observe $\vec{w}_1$ is the lab frame observer, then the interpolation degenerates to $\vec{w}(t;\alpha) = \big(\alpha\, a_2(t)\big)\, \vec{e}_1 + \big(\alpha\, b_2(t)\big)\, \vec{e}_2 + \big(\alpha\, c_2(t)\big)\, \vec{e}_3 = \alpha\, \vec{w}_2$, which is a scaling operation. See Fig. 4.4 for an example, where we directly interpolate between the (input) lab frame observer (given by the observer field $\vec{u} = 0$) and an observer "co-moving" along with the vortices.

Due to the linearity of the Lie algebra of Killing fields, this works for both $M = \mathbb{R}^2$ and $M = \mathbb{S}^2$, by using the corresponding basis $\{\vec{e}_i\}$.

4.4.3 Observer Interpolation Along a Path

Given an arbitrary observer field $\vec{u}$, we can pick two points $x_1(t) \in M$ and $x_2(t) \in M$, respectively, for an arbitrarily chosen fixed time t. We can then extract observers corresponding to any interpolated point $x(t;\alpha) \in M$, in between the two points. For simplicity, for $M = \mathbb{R}^2$ we connect the two points by a line segment, obtaining

$$x(t;\alpha) = (1-\alpha)\, x_1(t) + \alpha\, x_2(t), \quad 0 \leq \alpha \leq 1. \tag{4.20}$$

For $M = \mathbb{S}^2$, the linear connection between the two points does not exist, and we therefore use the geodesic arc between $x_1(t)$ and $x_2(t)$ instead.

4.5 Observer-Relative Quantities

In our framework, we want to be able to interactively change observer and extract features from the relative input field, i.e., the observed input vector field $\vec{v}$ and its derivatives, relative to the rigid motion described by $\vec{w}$. In terms of the observer $\vec{w}$ given by the function $t \mapsto (a(t), b(t), c(t))$, below we will make use of the following expansions and derivatives,

$$\begin{aligned}
\vec{w}(x,t) &= a(t)\,\vec{e}_1(x) + b(t)\,\vec{e}_2(x) + c(t)\,\vec{e}_3(x), \\
\nabla\vec{w}(x,t) &= a(t)\,\nabla\vec{e}_1(x) + b(t)\,\nabla\vec{e}_2(x) + c(t)\,\nabla\vec{e}_3(x), \\
\frac{\partial\vec{w}(x,t)}{\partial t} &= \frac{da(t)}{dt}\vec{e}_1(x) + \frac{db(t)}{dt}\vec{e}_2(x) + \frac{dc(t)}{dt}\vec{e}_3(x).
\end{aligned} \tag{4.21}$$

4.5.1 Reference Frame Transformation

In order to transform between two reference frames, we define a time-dependent diffeomorphism $t \mapsto \phi_t$, with each diffeomorphism, for fixed t, mapping on the same manifold M, i.e., $\phi_t \colon M \to M$. For $M = \mathbb{R}^2$, the isometry ϕ_t can be written explicitly

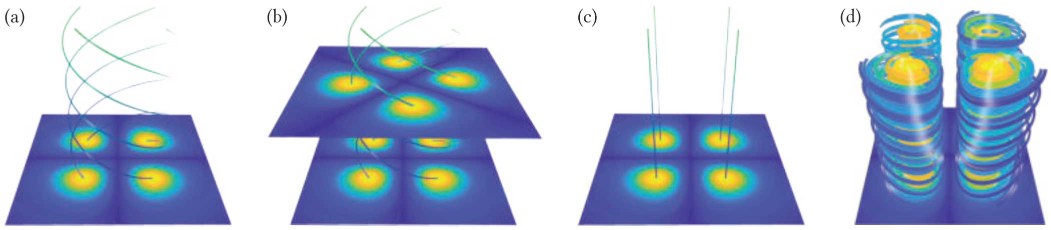


Figure 4.5: **Four centers**. (a,b) LAVD [1] and λ_2 vortex core lines agree (vertical axis is time). (c) Vortex core lines observed relative to an observer $\vec{w}$ selected from an observer field $\vec{u}$ computed via optimization [2]. (d) Observed path lines swirling around the core lines using variance color-coding.

as

$$\phi_t(x) = w(t) + \mathbf{Q}(t)\big(x - w(t_0)\big), \tag{4.22}$$

where $w(t_0)$ is some arbitrary position at $t = t_0$, and each $\mathbf{Q}(t)$ is a rotation tensor. In Cartesian coordinates, the rotation $\mathbf{Q}(t)$ is given by

$$\mathbf{Q}(t) = \begin{bmatrix} \cos\theta(t) & -\sin\theta(t) \\ \sin\theta(t) & \cos\theta(t) \end{bmatrix}. \tag{4.23}$$

The path $t \mapsto w(t)$ and the integrated angle $\theta(t)$ are the solutions of

$$\frac{d}{dt}w(t) = \vec{w}\big(w(t), t\big), \qquad \frac{d}{dt}\theta(t) = c(t), \tag{4.24}$$

with $\theta(t_0) = 0$. We can solve these two ODEs explicitly by

$$w(t) = w(t_0) + \int_{t_0}^{t} \vec{w}\big(w(\tau), \tau\big)\, d\tau, \qquad \theta(t) = \int_{t_0}^{t} c(\tau)\, d\tau. \tag{4.25}$$

To transform vectors and tensors with the diffeomorphism ϕ_t, we use the **pullback** ϕ_t^*. On $M = \mathbb{R}^2$, the pullback $\phi_t^* \, \vec{x}$ of a vector field $\vec{x}$ is

$$\phi_t^* \, \vec{x} = \mathbf{Q}^T(t) \, \vec{x}. \tag{4.26}$$

The pullback $\phi_t^* \, \mathbf{T}$ of a second-order tensor field $\mathbf{T}$ (a linear map) is

$$\phi_t^* \, \mathbf{T} = \mathbf{Q}^T(t) \, \mathbf{T} \, \mathbf{Q}(t). \tag{4.27}$$

For details on diffeomorphisms and pullbacks, and explicit formulas for $M = \mathbb{S}^2$, please refer to Appendices B and C.

4.5.2 Observed Velocity

The observed velocity, with respect to the observer $\vec{w}$, is obtained as

$$\vec{v}^* = \phi_t^* \left(\vec{v} - \vec{w} \right). \tag{4.28}$$

This velocity is of particular importance for vortex detectors that search for critical points of the velocity field, i.e., in this case the critical points of $\vec{v}^*$. In terms of the observer $\vec{w}$ as $t \mapsto (a(t), b(t), c(t))$, this becomes

$$\vec{v}^* = \phi_t^* \big(\vec{v} - a(t) \, \vec{e}_1 - b(t) \, \vec{e}_2 - c(t) \, \vec{e}_3 \big). \tag{4.29}$$

4.5.3 Observed Velocity Gradient

The observed velocity gradient, with respect to the observer $\vec{w}$, is

$$\nabla \vec{v}^* = \phi_t^* \big(\nabla \vec{v} - \nabla \vec{w} \big). \tag{4.30}$$

This velocity gradient is of particular importance for all vortex detectors that compute its eigenvalues and determine vortex-like behavior only when there are complex eigenvalues (2D: one conjugate complex pair). For these criteria, the relevant eigenvalues are therefore those of $\nabla\vec{v}^{*}$.

In terms of the observer $\vec{w}$ as $t \mapsto (a(t), b(t), c(t))$, this becomes

$$\nabla\vec{v}^{*} = \phi_t^{*}\Big(\nabla\vec{v} - a(t)\,\nabla\vec{e}_1 - b(t)\,\nabla\vec{e}_2 - c(t)\,\nabla\vec{e}_3\Big). \tag{4.31}$$

4.5.4 Observed Time Derivative

The observed time derivative [2, 3], with respect to the observer $\vec{w}$, is

$$\begin{aligned}\frac{\partial\vec{v}^{*}}{\partial t} &= \phi_t^{*}\left(\frac{\mathcal{D}}{\mathcal{D}t}(\vec{v}-\vec{w})\right),\\ &= \phi_t^{*}\left(\frac{\partial\vec{v}}{\partial t} - \frac{\partial\vec{w}}{\partial t} + \nabla\vec{v}\,(\vec{w}) - \nabla\vec{w}\,(\vec{v})\right).\end{aligned} \tag{4.32}$$

In terms of the observer $\vec{w}$ as $t \mapsto (a(t), b(t), c(t))$, this becomes

$$\begin{aligned}\frac{\partial\vec{v}^{*}}{\partial t} = \phi_t^{*}\Big(&\frac{\partial\vec{v}}{\partial t} - \frac{da(t)}{dt}\vec{e}_1 - \frac{db(t)}{dt}\vec{e}_2 - \frac{dc(t)}{dt}\vec{e}_3 +\\ &\nabla\vec{v}\big(a(t)\,\vec{e}_1 + b(t)\,\vec{e}_2 + c(t)\,\vec{e}_3\big) -\\ &\big(a(t)\,\nabla\vec{e}_1 + b(t)\,\nabla\vec{e}_2 + c(t)\,\nabla\vec{e}_3\big)\,\vec{v}\Big).\end{aligned} \tag{4.33}$$

See previous work [2, 3] for more on observed time derivatives.

4.6 Determining Observers from Prior Methods

We now demonstrate the generality and versatility of our framework by describing how observers can be defined via (1) Extraction from a globally optimized, continuous field of observers [2, 3]; (2) Extraction from separate observers optimized at each space-time point [7]; (3) Replication of Lagrangian-Averaged Vorticity Deviation (LAVD) [1] in our framework, relative to a special objectively-determined observer.

4.6.1 Observer Fields from Global Optimization

A directly suitable choice for the observer field $\vec{u}$ required by our framework are the observer fields computed by global optimization methods in the plane [2], or on the sphere [3], respectively. We have used this approach in the results shown in Fig. 6.1, Fig. 4.5, and Fig. 6.3.

4.6.2 Observer Fields from Generic Objective Vortices

In order to obtain a valid observer field from the **generic objective vortices** method of Günther et al. [7], we first use their optimization method to compute an optimal, objective field $\bar{\vec{v}}$ (Eq. 20 in [7]). From the objective field $\bar{\vec{v}}$, we then compute a smooth observer field $\vec{u}$ as

$$\vec{u} := \vec{v} - \bar{\vec{v}}. \tag{4.34}$$

This simple approach allows us to directly use the objective optimization method of Günther et al. [7] as an input to our framework. We have computed the observer field used in Fig. 1.1 with this approach.

We note that the above approach is objective and can indeed be used to obtain a valid observer field $\vec{u}$, despite the fact that the optimization [7] computes a **different** observer for each point in space-time. Since the objective vector field $\bar{\vec{v}}$ is defined at each space-time point, the observer field obtained via Eq. 4.34 describes valid, smooth observer motions. In fact, these motions are **determined objectively**, because the vector $\bar{\vec{v}}$ at each point in space-time is objective [7], and thus the corresponding velocities $\vec{u}$ are determined objectively. The obtained field $\vec{u}$ then enables the extraction of observers using our framework. We note, however, that, in general, we do not extract the exact same observers that are used for vortex detection in [7], because the derivatives given in [7] do not correspond to the field $\vec{u}$ that we obtain via Eq. 4.34.

4.6.3 Observers for Lagrangian-Averaged Vorticity Deviation

Our framework can be used to compute results equivalent to the LAVD method by Haller et al. [1], in the following way. (A detailed definition of LAVD will be given in section 6.1.2.)

As shown in Fig. 4.5, we define an observer $\vec{w}$, for $M = \mathbb{R}^2$, via $t \mapsto (a(t), b(t), c(t))$, by setting $a(t) = b(t) = 0$, and defining $c(t) := \bar{\omega}(t)$. Here, we set the angular velocity $\bar{\omega}(t)$ to half the average vorticity magnitude[2] of a chosen region $U(t) \subset \mathbb{R}^2$ of the flow field $\vec{v}$, as defined by Haller et al. [1]. We assume that the region $U(t)$ is invariant under the flow of $\vec{v}$, i.e., the region is deformed over time by the flow (map) of $\vec{v}$. We can then obtain the same scalar field as the LAVD field defined by Haller et al. [1], by simply integrating the **observed vorticity magnitude**, as observed by the observer $\vec{w}$, along path lines of the field $\vec{v}$ in the same region $U(t)$ used for computing $\bar{\omega}(t)$. We can obtain the observed vorticity from the **observed velocity gradient** $\nabla\vec{v}^*$ (Sec. 4.5.3), given by Eqs. 4.30, 4.31, giving the observed vorticity tensor as

$$\mathbf{\Omega}^* = \frac{1}{2}\left(\nabla\vec{v}^* - (\nabla\vec{v}^*)^T\right). \tag{4.35}$$

We note that due to our definition of the observer $\vec{w}$, the observed vorticity magnitude is identical to the instantaneous vorticity deviation (IVD) of Haller et al. [1], and therefore its integral along path lines is identical to LAVD. However, we do not need to explicitly compute the deviation of vorticity from the average vorticity, as Haller

[2] We note that the different definitions of angular velocity, vorticity (curl), and the vorticity tensor, require divisions or multiplications by two for conversion.

et al. [1] do: The vorticity that we integrate is simply the regular vorticity magnitude observed by $\vec{w}$. Nevertheless, the resulting LAVD scalar field is the same, because the observer $\vec{w}$ is rotating with the average angular velocity $\bar{\omega}(t)$, as defined by Haller et al. [1]. In this way, we gain additional insight into the meaning of LAVD: It is, in fact, the (integrated) vorticity magnitude seen by **a particular, objectively-determined observer**.

Averaging observers. We can gain even more insight from the following formulation: Given the region $U(t)$, we extract an observer from the field $\vec{u} := \vec{v}$ at every point $x \in U(t)$, for fixed t, obtaining a function $t \mapsto (a_i(t), b_i(t), c_i(t))$ for each $i = x$. We then define the **average observer** $\bar{\vec{w}}$ over $U(t)$ via Eq. 4.18, i.e., we average (a_i, b_i, c_i), over all $i = x \in U(t)$ to obtain $t \mapsto (\bar{a}(t), \bar{b}(t), \bar{c}(t))$. If we **project** this average observer to $t \mapsto (0, 0, \bar{c}(t))$, this is the same observer that we defined above via the average angular velocity $\bar{\omega}(t)$. IVD and LAVD can thus be formulated using the average observer of the region $U(t)$, followed by an appropriate projection. For our basis fields $\{\vec{e}_1, \vec{e}_2, \vec{e}_3\}$, $\bar{c}\,\vec{e}_3$ contains the entire vorticity. We can thus either project to $\vec{e}_3$, or notice that Eq. 4.35 is, in fact, independent of $\vec{e}_1$ and $\vec{e}_2$.

Chapter 5

System Design and Implementation

5.1 Interactive Visualization Pipeline

Fig. 5.1 shows the pipeline of our approach. The input to our system are two vector fields, i.e., an input field $\vec{v}$ and an observer field $\vec{u}$. The observer field can either come from an objective optimization provided by previous methods [7, 2, 3, 48] or we use the input field $\vec{v}$ as the observer field $\vec{u}$.

A user picks an observer in the GUI from the observer field. A set of path lines in the lab frame is computed in parallel. The resulting vertex positions are uploaded to a GPU buffer.

Using the numerical solution for world line for this observer, the time-dependent Killing field is computed and represented as $t \mapsto ((a(t), b(t), c(t))$ (Eq. 4.7). This representation enables further operations on the observer, e.g, interpolation, scaling, averaging.

After an observer is specified, we compute the time-dependent diffeomorphisms following Eq. 4.25 (reference frame transformations) using numerical integration (exponential map) on the CPU. We upload the resulting array of transformation matrices $\mathbf{Q}(t)$ for each time sample $t = t_i$ to the GPU buffer. Then, we apply the time-dependent diffeomorphism (transformation) to the vertexes of the path lines computed for the lab frame to get the relative visualization results. Each vertex is transformed by the corresponding transformation matrix (determined by the time component of this vertex) in the OpenGL vertex shader.

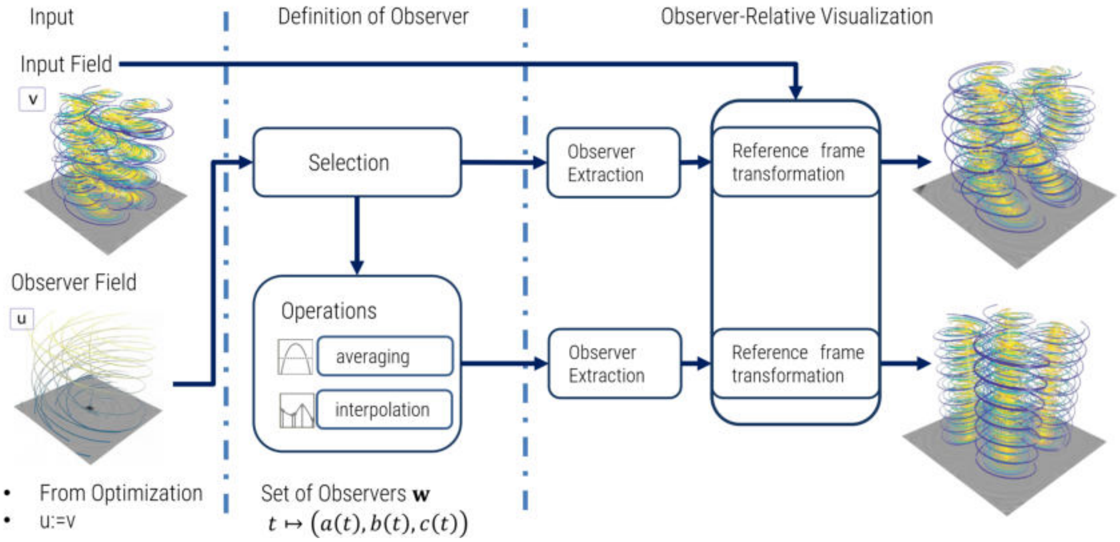


Figure 5.1: **Interactive visualization pipeline**. After user determine a new observer(select, load, modify a observer), the system first integrate observer's word line while extract corresponding time-dependent killing vector field at the same time; then compute the time-dependent diffeomorphism and apply it to stream objects(path lines and streamlines).

The computation is separated into three parts, and this separation allows us to independently update the set of path lines, as well as the observer, without recomputing the whole pipeline, e.g., when we change an observer, the path lines in the lab frame are not recomputed. As a result, this pipeline is very efficient and enables users to interactively change observers.

5.2 Framework Design and Implementation

Our framework is mostly implemented in C++ and OpenGL, for acceleration, CUDA is also used for paralleled path line integration. When users click the mouse to select seeds on the manifold, the Intel.Embree [58] library is used for ray-geometry intersections.

The overview of our framework is shown in Fig. 5.2(a). We can split the framework into three layers:

- Graphical User Interface (GUI) layer

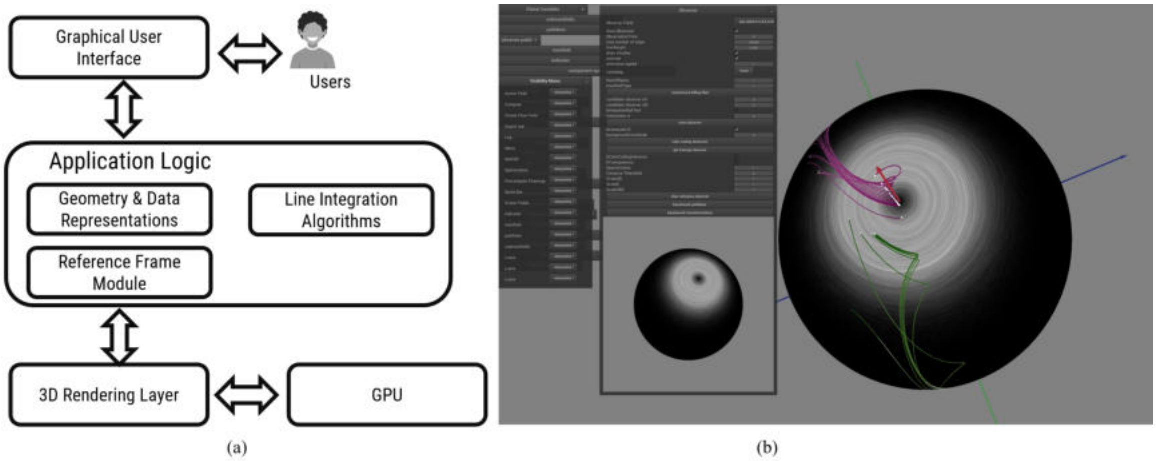


Figure 5.2: **System Overview.** Figure (a) shows the organization of the framework, figure (b) shows the User Interface.

- Application Logic

- 3D Rendering Layer.

The GUI layer is based on the ”nanogui” framework provided by Wenzel Jakob. Nanogui is a very lightweight and cross-platform widget library for OpenGL. We create multiple widget windows for users to control all the parameters and to trigger operations. Fig. 5.2(b) shows what the final Graphical User Interface Looks like.

The application logic layer contains the major parts of this novel technique [59], including data structures for the representation of geometry and flow data, as well as algorithms for path line integration, reference frame extraction, etc. We are using three flow visualization techniques to reveal the observer-relative features in the flow field: vector glyphs [16], line integral convolution (LIC) [10] and integral curves(streamlines and path line) [17, 18]. All the visualization techniques are implemented in the application logic layer, which can be categorized into three sections:

- Geometry and Data Representations: differential manifold implementation using charts, atlas,a triangle mesh data structure, velocity and scalar field data structures.

- Line Integration Algorithms: algorithms to solve ordinary differential equations of streamlines and path lines.
- Reference Frame Module: algorithms to integrate observer world lines, extract observers' motions and integration of the reference frame transformations.

In order to integrate path lines and streamlines on a manifold, we use charts and an atlas that cover the manifold. For the Euclidean plane $M = \mathbb{R}^2$, this is trivial. For curved spaces like a spherical manifold $M = \mathbb{S}^2$, the data representation is more complicated. We will talk about how the spherical manifold is covered by 6 coordinate charts in our implementation in section 5.2.1.

The Rendering Layer, will send the geometry of the manifold and the path lines to the shader (GPU). This results in an interactive visualization system.

In our framework, the manifolds (Euclidean plane and sphere) are modeled as triangular meshes. Path lines are modeled as vertex lists. A geometry shader is used to generate a view-aligned triangle strip for each path line. Vector fields are stored at the vertices of the triangle mesh of the manifold. For LIC, the vector field data is first resampled to a texture atlas. The two time steps that are adjacent to the currently visualized time are uploaded to the GPU. A fragment shader that is aware of the chart transition maps computes the LIC texture. Users have control over sampling parameters like path line stepsize and texture resolution.

5.2.1 Coordinate Charts on Spherical Manifold

We are solving the ordinary differential equations of streamlines and path lines by numerical integration of the vector field on the manifold.

The vectors are elements of the tangent spaces of the manifold's tangent bundle. Integration is equivalent to moving a massless particle in the direction of vector $\vec{v}(x,t)$ of the field at a spatial position x at time t. The integration in curved spaces can be computed directly in the 2D charts.

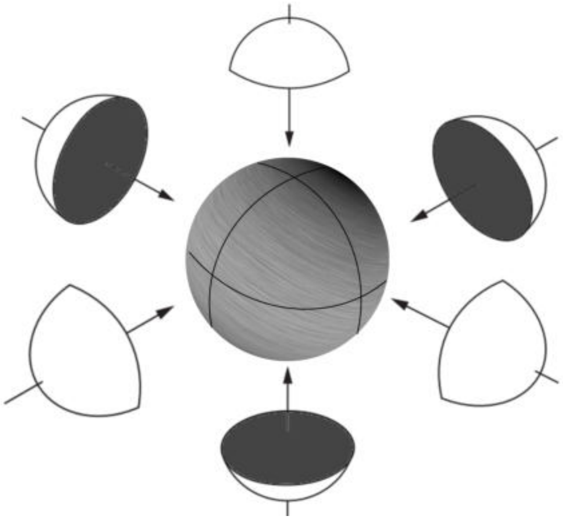


Figure 5.3: **Atlas for Sphere** $\mathbb{S}^2$. The six coordinate charts cover the whole sphere. When we compute integral curves on the sphere, we compute them in the 2D charts.

From the perspective of topology, the sphere is not homomorphic to the Euclidean plane, we need at least 2 charts to cover it. However, in our implementation, we use 6 coordinate charts to cover the sphere, as shown in Fig. 5.3. For a unit sphere $\mathbb{S}^2 = \{(x,y,z) \in \mathbb{R}^3; x^2+y^2+z^2=1\}$, we use the following parametrizations:

$$\begin{cases} x_1(x,y) = (x, y, +\sqrt{1-(x^2+y^2)}), & (x,y) \in U \\ x_2(x,y) = (x, y, -\sqrt{1-(x^2+y^2)}), & (x,y) \in U \\ x_3(x,z) = (x, +\sqrt{1-(x^2+z^2)}, z), & (x,z) \in U \\ x_4(x,z) = (x, -\sqrt{1-(x^2+z^2)}, z), & (x,z) \in U \\ x_5(y,z) = (+\sqrt{1-(y^2+z^2)}, y, z), & (y,z) \in U \\ x_6(y,z) = (-\sqrt{1-(y^2+z^2)}, y, z), & (y,z) \in U \end{cases} \tag{5.1}$$

where all the parametrizations are maps from $U \subset \mathbb{R}^2$ to $\mathbb{R}^3$, $U = \{(x,y) \in \mathbb{R}^2; x^2+y^2=1\}$. These parametrizations cover $\mathbb{S}^2$ completely (Fig. 5.3) and imply that $\mathbb{S}^2$ is a regular surface.

5.2.2 Integration of Path Lines and Stream Lines

Path lines and stream lines are the solutions of ordinary differential equations as shown in section 2.4. Algorithm 1 shows the implementation of the numerical solution for the ordinary differential equation of path lines.

Algorithm 1 shows the procedure to integrate path lines on a manifold in one time direction. We use the same algorithm to integrate forward and backward in time. To integrate backward we simply make the stepsize parameter negative.

When building the mesh data structure of the manifold, we compute the optimal chart for every triangle. When a triangle lies in just one chart, the optimal chart is trivial; when a triangle lies in the shared area of two or more charts, we use a heuristic to solve the ambiguity. The chart where the triangle is farthest from the boundary (for equal distance, with the smallest chart id) is selected as the optimal chart for each triangle.

The particle's chart id, chart coordinates and time information is stored as $p = (c, x, t)$, where c denotes the chart id, x denotes the coordinates in the chart c and t denotes the time. Every iteration of the integration process, we first check whether the particle's position is represented in the optimal chart. Otherwise, we use a map between charts to map it from the non-optimal chart to the optimal chart. This is done by: first using maps provided in equation 5.1 to map the particle into 3-dimensional space and then using the corresponding inverse map of the optimal chart to map it into the optimal chart.

After a seed point $p_0 = (c_0, x_0, t_0)$ was selected by the user, algorithm 1 will compute the corresponding path line for this seed. The seeding time t_0 can be freely chosen. The integration is performed twice: integration forward and backward in time. In practice, thousands of seed points can be selected, and the integration is computed in parallel on a multi-processor CPU or GPU (using CUDA). For path line integration in Euclidean space, the solution works analogously but the chart transitions can be omitted.

An observer's world line is a path line of the observer field $\vec{u}(x,t)$. The only difference is that when we integrate an observer's world line, we will store all the velocities and covariant derivatives along that world line in order to compute a time-dependent Killing field from it.

Algorithm 1 Path Line Integration on a Manifold

```
Input: Manifold $S$; Seed position $p_0 = (c_0, x_0, t_0)$; Vector field $\vec{v}$ on $S$, and its time
    domain [$T_{min}$, $T_{max}$]; Step size $\delta$;
Output: Path line (a list of positions) $Y = \{p^0, p^1, ...p^n\}$
initial  $p = p_0$, $Y = \{p_0\}$;
while $p.t \leq T_{max}$ and $p.t \geq T_{min}$ do
    Find optimal chart id $c_o(p)$ for $p$
    if $p.c \neq c_o(p)$ then
        Get the transition map $\varphi$ from chart $p.c$ to chart $c_o(p)$
        Map $p$ from chart $p.c$ to chart $c_o(p)$: $p = (c_o(p), \varphi(x), t)$
    end if
    Get velocity $\vec{v}_p = \vec{v}(p.c, p.x, p.t)$
    $p_p = (p.c, p.x + \vec{v}_p \cdot \delta, t + \delta)$
    Insert $p_p$ into $Y$
    Set $p = p_p$
end while
```

5.2.3 Reference Frame Transformation

We compute path lines in the input field relative to the lab reference frame. Based on the equations given in section 4.5, our framework carries out the computation of an observer's reference frame transformation. After we compute all time-dependent diffeomorphisms $t \mapsto \phi_t$ for all discrete time steps, we interpolate one diffeomorphism for each time step of the path lines. We then compute the exponential map for each t, which results in an array of matrices. The matrices are uploaded to the GPU and used by the vertex shader. The transformation of the path line vertices is computed interactively.

Reference Frame Transformation for Observed LIC is computed similarly: we transform the vector field to the relative vector field in the fragment shader, using the time-dependent diffeomorphisms, and integrate stream lines within the shader.

5.3 Visualization and Interaction

5.3.1 Observer Specification

We implemented an "observer" widget, through which the user can determine a new observer by clicking a seed point on the manifold.

The objective optimization methods [7, 2, 60, 3] can be configured from the GUI.

5.3.2 Observer Manipulation

The visual representation of the observer (a,b,c) components is visualized in an additional widget. An illustration of the GUI is shown in Fig. 5.4. Interpolation and averaging of observers are supported and can be specified in the GUI.

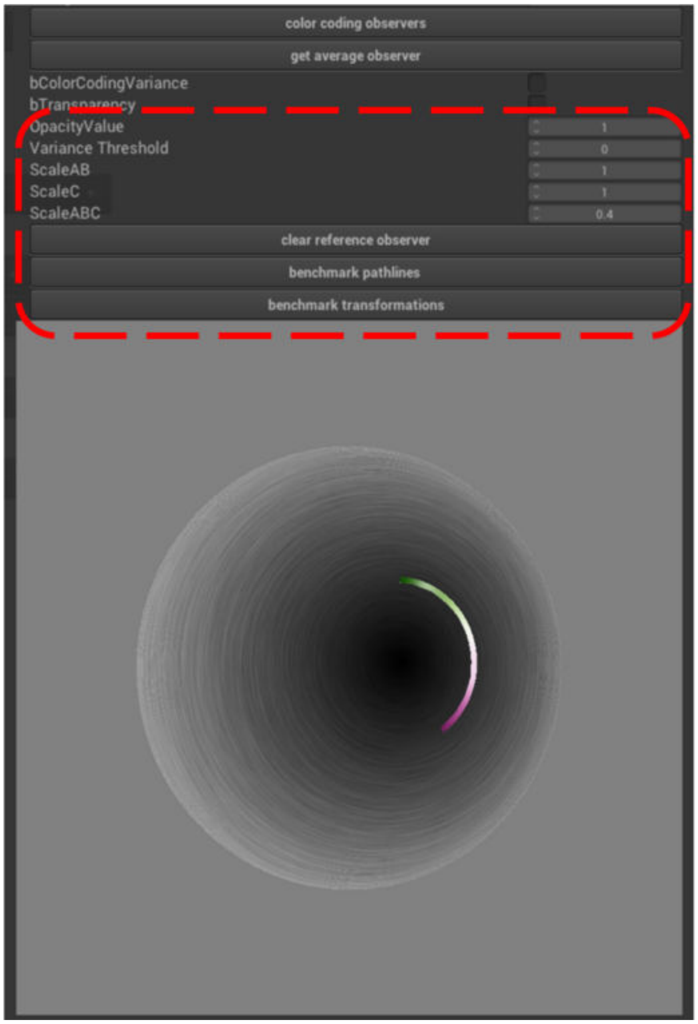
5.3.3 Observer Comparison and Exploration

We visualize differences between observers as a color coded scalar field on the manifold. For a chosen observer O_1, our framework shows a color coding of a set of observers on the manifold M.

Fig. 5.5 shows the color coding for two different observers. Fig. 5.5 (a) shows an example of a badly chosen observer. The user can compare all other candidate observers on the manifold and select a candidate observer that is different from previous observer. Fig. 5.5 (b) shows the visualization for a better chosen observer.

5.4 Performance Analysis

In our framework, the computation of observer relative visualization can be separated into three parts: the integration of path lines including the observer word line and



bColorCodingVariance	□
bTransparency	□
OpacityValue	1
Variance Threshold	0
ScaleAB	1
ScaleC	1
ScaleABC	0.4

Figure 5.4: **Observer manipulation**. An observer can be scaled, interpolated, or averaged from a predetermined set of observers.

path lines relative to lab frame in the input field; the computation of time-dependent diffeomorphisms (observer's reference frame transformation); and the transformation in the shader.

Our experiments are carried out on a workstation with two Intel Xeon 6230R processors with a total of 52 CPU cores, and an NVIDIA GeForce RTX 3090 GPU. We have run performance tests for different data sets, numbers of path lines, and integration step sizes.

Table 5.1 lists the time we spend for path line integration. All performance numbers were averaged over 1,000 runs of path line integration. The number of integra-

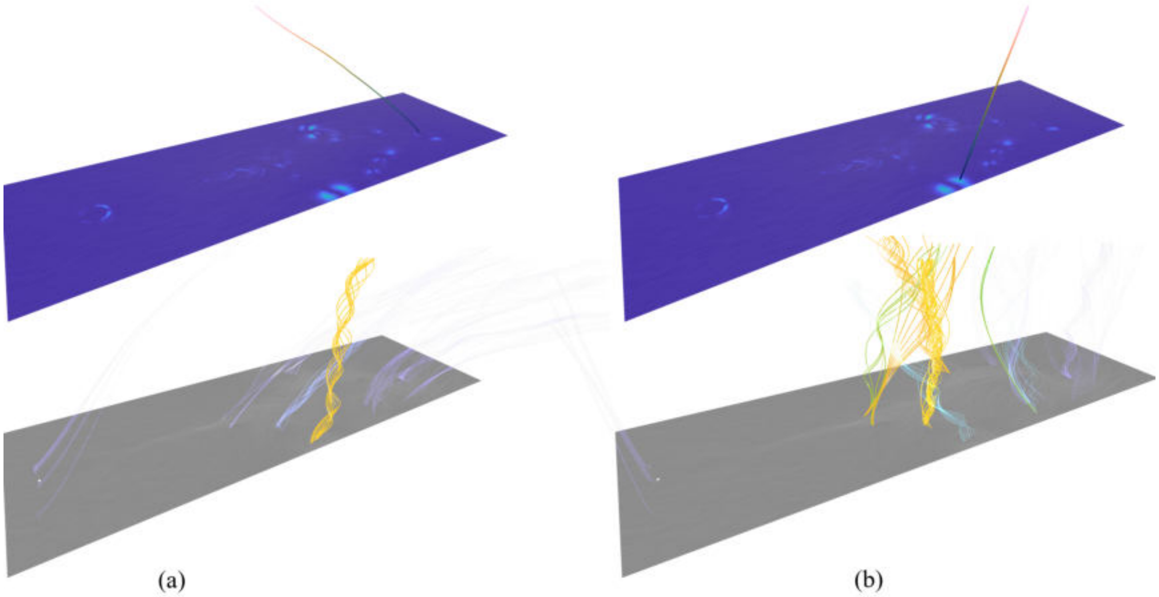


Figure 5.5: **Observer exploration**. The color maps (top figures) indicate the difference of other candidate observers on the manifold. Users can select new observers using the guidance of the color maps (selecting a new observer at bright spots in the color map). The corresponding observer-relative visualization is shown at the bottom.

Table 5.1: **Path line integration** performance.

data set	number of path lines	time [ms]	throughput [vertices/s] in millions
Four centers	64	26	15
	128	48	17
	256	83	19
	512	154	21
	1,024	238	27
Heated cylinder (Boussinesq)	64	23	14
	128	40	16
	256	76	17
	512	150	17
	1,024	293	17

tion steps that are required to reach high-quality visualizations strongly depends on the data set. We typically use two to ten samples for each time step of a data set. Table 5.1 shows that the performance mostly depends on the number of vertices generated by the numerical integration. When integrating enough path lines in parallel we achieve a throughput of more than 15 million vertices per second.

Similarly, table 5.2 shows performance numbers that were averaged over $1,000$

Table 5.2: **Diffeomorphism computation** performance.

data set	number of samples	time [ms]	throughput [samples / s] in thousands
Four centers	6.28×10^1	5	12
	6.28×10^2	6	100
	6.28×10^3	12	509
	6.28×10^4	87	723
Heated cylinder (Boussinesq)	5×10^1	11	4
	5×10^2	13	38
	5×10^3	21	234
	5×10^4	82	609

runs for the computation of the transformation matrices used for the diffeomorphism. At every point of the observer world line, we need to compute $\nabla \vec{w}$ and extract a Killing vector field. The performance depends on the number of time steps of the data set as well as the number of samples taken along the observer world line. In all our experiments, the computation of the diffeomorphism, with a high number of samples, including the transfer of the data to the GPU, takes less than 100 ms.

Applying the diffeomorphism in the vertex shader requires one additional matrix-vector multiplication per vertex. The performance impact of this step is negligible. We measured the frame rate of our rendering algorithm and found that we consistently get 30 frames per second for path line rendering as well as observed path line rendering.

Chapter 6

Experiments on Flow Data

Our framework can be used to interactively explore unsteady vector fields. It enables users to view the relative velocity field using a large number of meaningful observers. In the following, we showcase the application of our interactive framework to explore many classical flow field data sets and demonstrate the efficiency of the framework.

In particular, we explore the vector space corresponding to all physically-realizable observers together with different observer-relative transformed fields, and analyze the resulting perception of vortex structures. The data sets we use to demonstrate our framework come from both analytical test cases and numerically-simulated flow fields. The details about the data sets are given in table 6.1.

Table 6.1: **Data set** sizes used for evaluation.

Data set	Size	Time steps
Cylinder Flow with von Kármán vortex street (simulated)	640 × 80	1,000
Heated cylinder with Boussinesq approximation (simulated)	150 × 450	800
Beads problem (analytic)	128 × 128	512
Divergence free beads flow (analytic)	128 × 128	256
Four rotating centers (analytic)	128 × 128	256
Cylinder flow Jung, Tel and Ziemniak (analytic)	450 × 200	500
Bickley jet (analytic)	300 × 60	300
Vortex on sphere triangle mesh (analytic)	7,608 triangles	64

6.1 Vortex Detection Criterion

We evaluated our visualization framework and compared our visualizations with results from vortex detection criteria.

6.1.1 λ_2 Criterion

The λ_2 criterion proposed by Jeong and Hussein [44], considers the eigenvalues of the symmetric matrix $\mathbf{S}^2 + \Omega^2$. For a region to be considered as a vortex, the second largest eigenvalue λ_2 must be negative. Vortex regions are the areas satisfying the following equation:

$$\lambda_2(\mathbf{S}^2 + \Omega^2) < 0 \tag{6.1}$$

where $\mathbf{S}$ and Ω is the decomposition of the velocity gradient tensor.

6.1.2 Lagrangian-Averaged Vorticity Deviation (LAVD)

Lagrangian-Averaged Vorticity Deviation (LAVD) proposed by Haller et al. [1] is an objective criterion relative to a specific global observer. We consider an unsteady velocity field $\vec{v}(x,t)$, defined on a possibly time-dependent spatial domain $U(t) \in \mathbb{R}^3$ over a finite time interval $[t_0, t_1]$. The instantaneous spatial mean vorticity over $U(t) \in \mathbb{R}^3$ is defined as:

$$\bar{\omega}(t) = \frac{\int_{U(t)} \vec{\omega}(x,t)dV}{vol(U(t))} \tag{6.2}$$

where $\vec{\omega}(x,t)$ is the vorticity at position x and time t, $vol(\cdot)$ denotes the volume for three-dimensional flows and the area for two-dimensional flows.

The particles will move in the flow field, and we can represent the trajectories of particles by defining the flow map:

$$\psi_{t_0}^{t} : x_0 \longmapsto x(t; x_0), t \in [t_0, t_1] \tag{6.3}$$

as the mapping from initial particle positions $x_0 \in U(t)$ to their later positions $x(t; x_0) \in U(t)$.

Finally, the LAVD from time t_0 to time t is defined as:

$$LAVD_{t_0}^{t} = \int_{t_0}^{t} |\omega(x(s; x_0), s) - \bar{\omega}(s)| ds \tag{6.4}$$

6.2 Four Centers

This analytical 2D vector field on $M = \mathbb{R}^2$ contains four vortices swirling around four vortex centers. The four centers are rotating around a common axis orthogonal to the 2D plane, Fig. 4.5 shows a visualization of the data set.

It is constructed from a steady flow field given by,

$$\vec{v}(x, y) = \begin{pmatrix} -x(2y^2 - 1)e^{(-(x^2+y^2))} \\ -x(2x^2 - 1)e^{(-(x^2+y^2))} \end{pmatrix}. \tag{6.5}$$

The steady vector field is defined in domain $\mathbb{D} = [-2, 2] \times [-2, 2]$. In order to get the final unsteady flow data, the steady field is observed from an observer that rotates with unit speed around the axis, which passes the origin and is orthogonal to the 2D plane.

In Figs. 4.5 (a,b), we show that LAVD [1] (color-coded) detects these four vortices, and agrees with the result of the λ_2 criterion [44]. Fig. 4.5(c) shows the observed vortex core lines relative to an observer that rotates along these four vortices, as a result, the relative visualization of path lines are straight lines. We have computed this observer via global optimization [2]. The observed vortex core lines stay in place. In this reference frame, it is now trivial to verify that path lines seeded around the centers swirl around the vortex cores (Fig. 4.5(d)). The color coding of the swirling path lines in Fig. 4.5(d) is computed according to path line variance. The vertical axis of the path lines encodes the time.

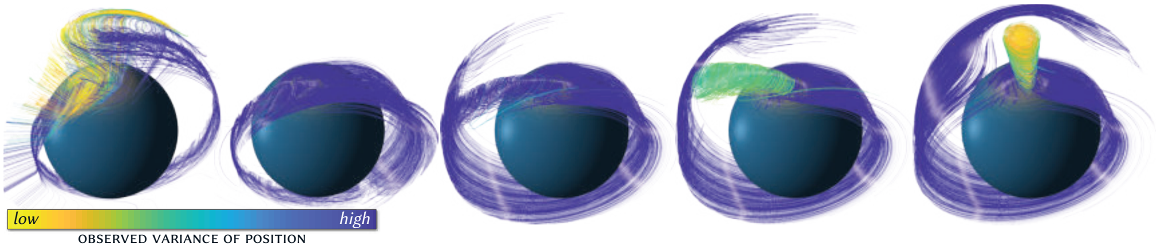


Figure 6.1: **Vortex on the sphere.** We exploit the 3D vector space structure of the Lie algebra of observer motions on spherical domains. Here, we smoothly interpolate between two observers, the input lab frame (left) and an objectively-determined observer [3] (right), smoothly "shifting" the observed field and the contained vortex until the latter becomes visible (right). The axis orthogonal to the sphere corresponds to time.

6.3 Vortex on the Sphere

Our framework works on Euclidean spaces as well as curved spaces like the spherical domain. We construct an analytical steady vector field on the unit sphere that contains a vortex. The unsteady data set is generated by using a reference frame movement once around the sphere.

In Fig. 6.1 (left), the path lines observed by the lab frame are hard to interpret. Fig. 6.1 shows snapshots of the observer relative transformations interpolated (left to right) between the lab frame and a globally optimized observer [3] that moves with the vortex. The last frame in the sequence clearly shows the vortex that is observed by the objectively-determined, co-moving observer.

6.4 Cylinder Flow

The Cylinder flow (Jung et al. [61]) is used to study the flow of an inviscid, incompressible fluid around an obstructing cylinder that is transverse to the flow. The vortices behind the cylinder are of interest. These vortices are short-lived, and new vortices develop periodically. Thus, these vortices are hard to see.

In Fig. 4.4, the vertical axis and the color map encode time. We carry out interpolation (Fig. 4.4, a-d) between the lab frame observer and an objectively-determined constant-velocity observer that co-moves with the vortices, revealing the hidden and short-lived vortices. We demonstrate that a moving observer helps to produce a less obstructed view of the individual vortices.

By animating the observer's translation, we provide a visual clue how the data observed in the lab frame is transformed into the frame of a moving observer. The transition between the lab frame and the final observer results from interpolation (Sec. 4.4.2).

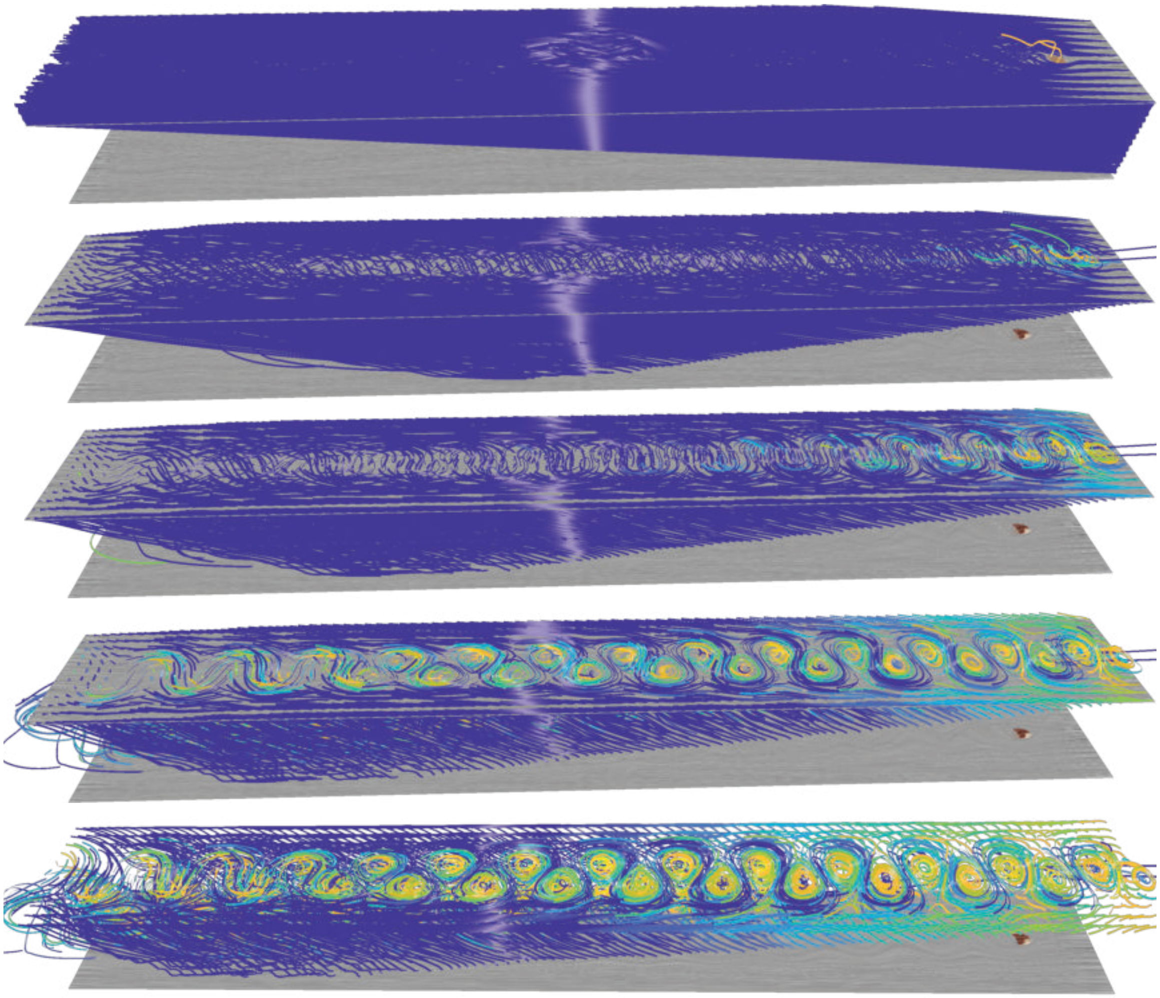


Figure 6.2: **Vortex street (cylinder).** Changing the observer interactively (top to bottom) allows revealing the vortices (bottom) obstructed in the lab frame (top). (The vertical axis in each visualization corresponds to time.)

6.5 Vortex Street (Cylinder)

This numerical simulation data describes the flow behavior of a vortex street behind a cylindrical obstacle, which is easier to interpret in a moving reference frame. The data is shown in Fig. 6.2.

Unlike other examples, we have not computed an observer using objective optimization. In this example, we set the observer field $\vec{u}$ equal to the input field $\vec{v}$. A suitable co-moving observer is picked from the ambient flow of the input field. The illustration shows that extracting an observer from the ambient flow of the input field $\vec{v}$ can be good enough to define a suitable observer in some cases.

Fig. 6.2 shows snapshots of the interpolation between the lab frame visualization (top) to the co-moving observer from the input field $\vec{v}$ (bottom).

We point out that, although the observer was extracted along a path line in the input flow, our mathematical framework guarantees that each individual frame of the sequence corresponds to a physically-observable visualization, relative to a rigid motion observer. We automatically compute a color-coding based on the **variance** of the spatial locations (relative to the chosen observer) of a particle over time, along its path line. This color-coding highlights the centers of vortices, provided that the chosen observer perceives the particle as swirling around a mostly static center. Low variance is mapped to brighter hues in the color map. This effect is clearly visible in Fig. 6.2 (bottom).

6.6 Bickley Jet

This data set (Fig. 6.3) [1] contains an analytically-defined vortex street. Relative to a moving observer, the vortex structures become clearly visible. We compute the observer field $\vec{u}$ via global optimization [2]. In Fig. 6.3, the vertical axis of the path lines encodes the time. Fig. 6.3(a) shows path lines in the lab frame, which are hard to interpret. Fig. 6.3(b) shows observed path lines, making the vortices better visible. We keep the color for each path line constant (minimum observed variance) for the transformation sequence (Fig. 6.3(a), (b)). This demonstrates that color-coding is not enough to show the vortex structures. The transformation of the path lines achieves a better visualization.

6.7 Beads Flow

The beads flow was described by Wiebel et al. [62]. We use the analytic version defined by Weinkauf and Theisel [18] and by Günther et al. [23]. The input flow field relative to the lab frame is shown in 6.4(a), which contains a contracting flow (a sink) that rotates around a center as observed by the input lab frame. Fig. 6.4(b) shows the flow relative to an observer that was selected from the input field $\vec{u} = \vec{v}$. We see that all observed path lines are now straight lines (depicted as “shadows”) that converge at the same point.

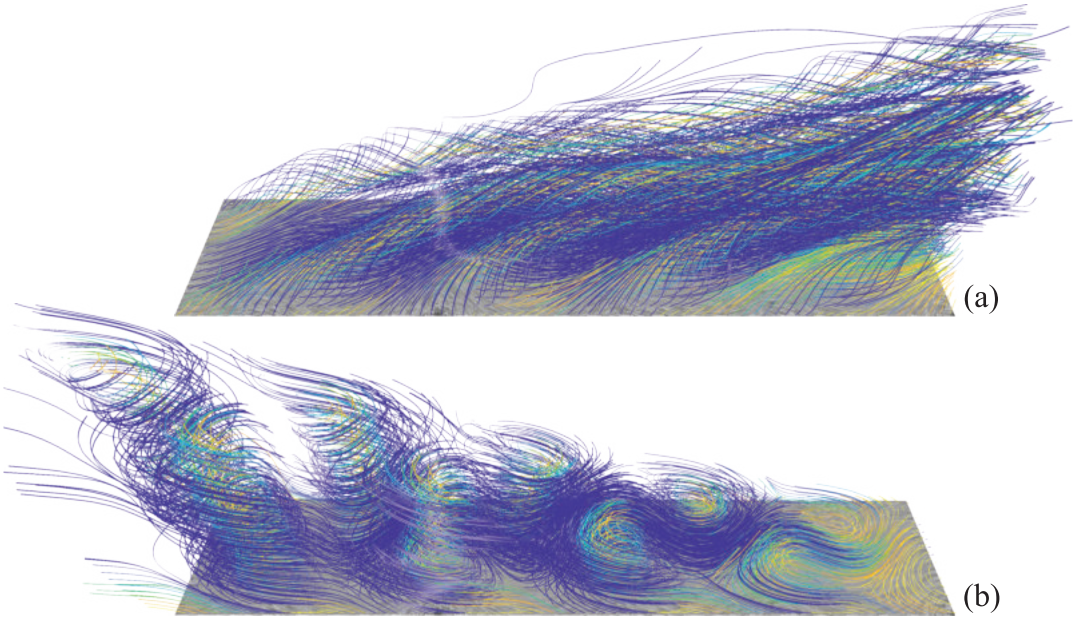


Figure 6.3: **Bickley jet**. The input lab frame obstructs the visibility of vortices (a). A globally optimized observer [2] reveals the vortices clearly (b). (The vertical axis in each visualization corresponds to time.)

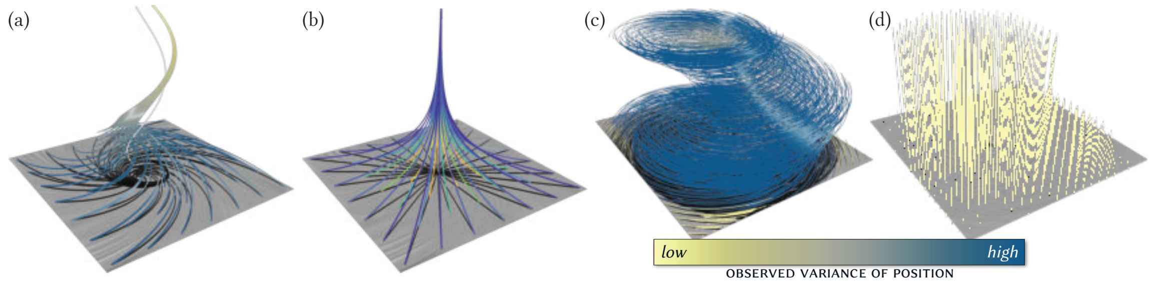


Figure 6.4: **Beads flow.** (Original; a, b) The input flow field (vertical axis is time) contains rotational as well as contracting motion (negative divergence). However, using our framework one can determine that all rotational parts of this field can be fully explained by a rotating reference frame. (Divergence-free; c, d) Observer exploration enables us to see that, in fact, this flow field is itself a Killing field. Thus, the input field (c) can be explained purely by a moving reference frame observing a 0-field (d). This indicates that this field does not contain any intrinsic motion, beyond the motion of an observer.

We can see that the input flow field contains rotational as well as contracting motion. However, using our framework one can determine that all rotational parts of this field can be fully explained by a rotating reference frame, and only contracting motion remains in Fig. 6.4(b). Using our framework we have found that all observers extracted from the input field $\vec{v}$ perceive this flow field without any rotational motion. They are, in fact, all rotating with the same angular velocity. Since we have objectively determined observers that perceive this flow without any rotational motion, we must conclude that all observed rotation in this flow is solely due to the rotation of a reference frame, and the flow itself does not contain intrinsic rotational motion. We note, however, that this is a purely **kinematic** judgement. In terms of **dynamics**, a rotating frame and an inertial frame behave differently.

6.8 Divergence-free Beads Flow

This vector field is a divergence-free version of the original beads flow data [23], used as a test case for rotation-invariant vortex detection. Using our interactive observer exploration framework, we can see that the input flow in Fig. 6.4(c) is in fact itself a Killing field: We choose the input field as the observer field, i.e., $\vec{u} := \vec{v}$, and interactively see that all observers extracted from this observer field are in fact the same rigid-body observer. The observed field is zero, as shown in Fig. 6.4(d). Since the motion described by this vector field is purely explainable by an observer motion, the input field contains no intrinsic motion. The same is confirmed by an observer field computed via global optimization [2]: The optimized observer field $\vec{u}$ is $\vec{u} = \vec{v}$.

A fundamental problem with detecting a vortex in this data set is that, since the field can be observed as the 0-field, any rotating observer can be used in order to perceive a vortex at any chosen location. The only consistent interpretation for this problem is that this vector field does not contain objective vortices.

6.9 Boussinesq

This data set (shown in Fig. 1.1) demonstrates the need for an interactive exploration tool, because vortices are moving in different directions. We first conduct the objective optimization method by Günther et al. [7] for the Boussinesq flow field, obtaining an observer field $\vec{u}(x, t)$ on the manifold M as described in section. 4.6.2. We then visualize the flow relative to one physically-realizable observer at a time. We smoothly transition between a large number of observers interactively selected by the user, by changing the location where an observer is extracted from the observer field. Fig. 1.1(a) depicts six different observer-relative visualizations. More snapshots of this example is shown in Fig 6.5. We modulate the opacity of each path line with its variance; path lines with high variance become fully transparent, making vortex structures fade in and out while the observer is being changed, as demonstrated in the six sub-figures and also in our videos (see Appendix for Supplementary Materials).

Figs. 1.1(b, c) show interactive tools provided by our framework to help guide the exploration of the vast space of possible observers. Fig. 1.1(b) shows the (a, b, c) parameter space of linear velocity (horizontal) and vorticity (vertical; positive in red, negative in blue). This provides further insight into the space of possible observers: Each observer is a curve in the 3D parameter space; the six observers from Fig. 1.1(a) correspond to six curves in Fig. 1.1(b). Fig. 1.1(c) depicts color-coded observer similarity (Eq. 4.10), which can help identify observers that are significantly different from the current observer.

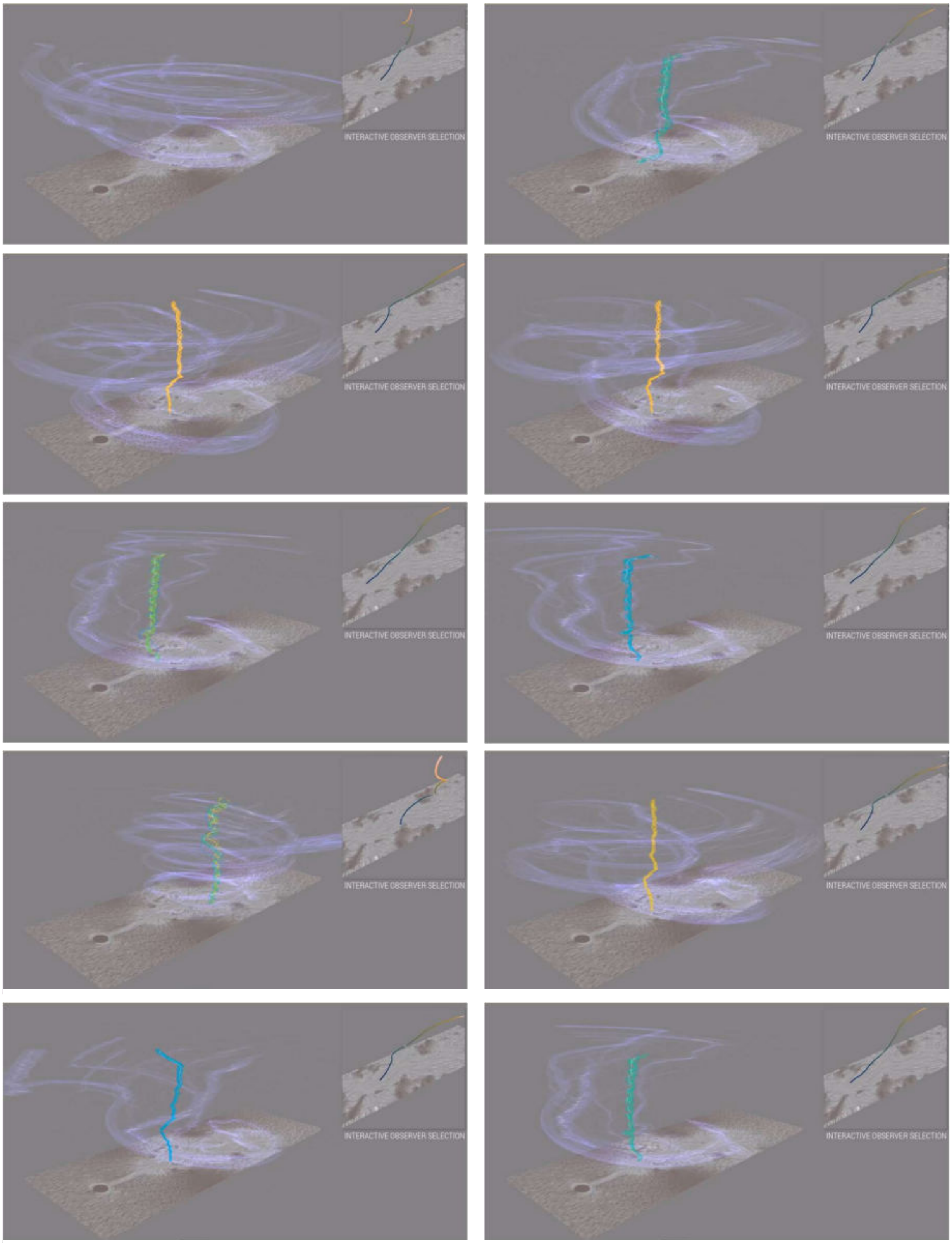

INTERACTIVE OBSERVER SELECTION
INTERACTIVE OBSERVER SELECTION
INTERACTIVE OBSERVER SELECTION
INTERACTIVE OBSERVER SELECTION
INTERACTIVE OBSERVER SELECTION
INTERACTIVE OBSERVER SELECTION
INTERACTIVE OBSERVER SELECTION
INTERACTIVE OBSERVER SELECTION
INTERACTIVE OBSERVER SELECTION
INTERACTIVE OBSERVER SELECTION

Figure 6.5: **Exploring the Boussinesq flow.** Path lines are variance based color-coded, vertical axis encodes time. Small insets depict observer world lines.

Chapter 7

Concluding Remarks

7.1 Summary

Our framework demonstrates that for revealing vortex structures in flow fields where one reference frame is not sufficient, interactive exploration of the space of observers is necessary. Our framework is flexible enough to incorporate observers that are objectively computed with several different state-of-the-art methods. In contrast to prior work computing features jointly, we focus on the individual, physically-realizable observers for visualization and vortex detection. We believe that our framework can provide a common basis for the future investigation of the interaction between feature detectors derived using physical arguments on the one hand, and the flexible definition of observer fields on the other.

Determining observers objectively can be crucial. For example, our exploration of the beads flow has shown that when observers are not chosen objectively, the perceived rotation can be due to the choice of observer: Rotation can be detected where there is no intrinsic (kinematic) rotation. We have also illustrated several different ways of choosing observers objectively. We therefore believe that further investigation of the meaning and implications of objective vortex detection and objectively determining observers is necessary, and we hope that our framework can contribute to this endeavor.

7.2 Future Research Work

Our current implementation works on two-dimensional manifolds. The mathematical framework also works on three-dimensional manifolds. An extension of the software to 3D is planned. It is also interesting to explore the parameter space of the physically-realizable observers' representation, as any observer can be represented as a curve $t \mapsto (a(t), b(t), c(t))$ in this parameter space. Besides, it will be interesting to see how the framework applies to features that exhibit drastic dynamic motion in more complex flow data sets.

REFERENCES

[1] G. Haller, A. Hadjighasem, M. Farazmand, and F. Huhn, “Defining coherent vortices objectively from the vorticity,” **Journal of Fluid Mechanics**, vol. 795, pp. 136–173, 2016.

[2] 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**, vol. 25, no. 1, pp. 1257–1266, 2019.

[3] 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**, vol. 27, no. 2, pp. 283–293, 2021.

[4] G. H. Shaker, “Interactive exploration of objective vortex structures in unsteady flow using observer fields,” Ph.D. dissertation, 2019.

[5] A. J. Smits, **Flow visualization: techniques and examples**. World Scientific, 2012.

[6] G. Haller, “An objective definition of a vortex,” **Journal of Fluid Mechanics**, vol. 525, pp. 1–26, 2005.

[7] T. Günther, M. Gross, and H. Theisel, “Generic objective vortices for flow visualization,” **ACM Transactions on Graphics**, vol. 36, no. 4, p. Article No. 141, 2017.

[8] T. Günther and H. Theisel, “The State of the Art in Vortex Extraction,” **Computer Graphics Forum**, vol. 37, no. 6, pp. 149–173, 2018.

[9] A. C. Telea, **Data visualization: principles and practice**. CRC Press, 2014.

[10] B. Cabral and L. C. Leedom, “Imaging vector fields using line integral convolution,” in **Proceedings of the 20th annual conference on Computer graphics and interactive techniques**, 1993, pp. 263–270.

[11] M. P. Do Carmo, **Differential geometry of curves and surfaces: revised and updated second edition**. Courier Dover Publications, 2016.

[12] T. Frankel, **The Geometry of Physics: An Introduction**, 3rd ed. Cambridge University Press, 2011.

[13] M. P. do Carmo, **Riemannian Geometry**. Birkhäuser, 1992.

[14] J. Stillwell, **Naive lie theory**. Springer Science & Business Media, 2008.

[15] C. Truesdell and W. Noll, **The Nonlinear Field Theories of Mechanics**. Springer-Verlag, 1965.

[16] R. Borgo, J. Kehrer, D. H. Chung, E. Maguire, R. S. Laramee, H. Hauser, M. Ward, and M. Chen, "Glyph-based visualization: Foundations, design guidelines, techniques and applications." in **Eurographics (State of the Art Reports)**, 2013, pp. 39–63.

[17] B. Jobard and W. Lefer, "Creating evenly-spaced streamlines of arbitrary density," in **8th Eurographics Workshop on Visualization in Scientific Computing**, 1997, pp. 211–222.

[18] T. Weinkauf and H. Theisel, "Streak lines as tangent curves of a derived vector field," **IEEE Transactions on Visualization and Computer Graphics**, vol. 16, no. 6, pp. 1225–1234, 2010.

[19] F. H. Post, B. Vrolijk, H. Hauser, R. S. Laramee, and H. Doleisch, "The state of the art in flow visualisation: Feature extraction and tracking," in **Computer Graphics Forum**, vol. 22, no. 4. Wiley Online Library, 2003, pp. 775–792.

[20] H. Bhatia, G. Norgard, V. Pascucci, and P.-T. Bremer, "The helmholtz-hodge decomposition—a survey," **IEEE Transactions on Visualization and Computer Graphics**, vol. 19, no. 8, pp. 1386–1404, 2012.

[21] 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**, vol. 13, no. 4, pp. 641–651, 2007.

[22] 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**, vol. 13, no. 6, pp. 1759–1766, 2007.

[23] T. Günther, M. Schulze, and H. Theisel, "Rotation invariant vortices for flow visualization," **IEEE Transactions on Visualization and Computer Graphics**, vol. 22, no. 1, pp. 817–826, 2016.

[24] G. A. Holzapfel, **Nonlinear Solid Mechanics: A Continuum Approach for Engineering**. Wiley, 2000.

[25] R. W. Ogden, **Non-Linear Elastic Deformations**. Dover Publications, Inc., 1997.

[26] 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**, vol. 28, no. 2, pp. 189–198, 1976.

[27] G. Astarita, “Objective and generally applicable criteria for flow classification,” **Journal of Non-Newtonian Fluid Mechanics**, vol. 6, no. 1, pp. 69–76, 1979.

[28] T. Günther and H. Theisel, “Objective vortex corelines of finite-sized objects in fluid flows,” **IEEE Transactions on Visualization and Computer Graphics**, vol. 25, no. 1, pp. 956–966, 2018.

[29] R. Bujack, M. Hlawitschka, and K. I. Joy, “Topology-inspired Galilean invariant vector field analysis,” in **Proceedings of IEEE Pacific Visualization 2016**, 2016, pp. 72–79.

[30] B. Kim and T. Günther, “Robust reference frame extraction from unsteady 2d vector fields with convolutional neural networks,” in **Computer Graphics Forum**, vol. 38, no. 3. Wiley Online Library, 2019, pp. 285–295.

[31] H. J. Lugt, “The dilemma of defining a vortex,” in **Recent Developments in Theoretical and Experimental Fluid Mechanics: Compressible and Incompressible Flows**. Springer-Verlag, 1979, pp. 309–321.

[32] S. K. Robinson, “A review of vortex structures and associated coherent motions in turbulent boundary layers,” in **Proceedings of Structure of Turbulence and Drag Reduction, IUTAM Symposium**. Springer-Verlag, 1990, pp. 23–50.

[33] D. C. Banks and B. A. Singer, “A predictor-corrector technique for visualizing unsteady flow,” **IEEE Transactions on Visualization and Computer Graphics**, vol. 1, no. 2, pp. 151–163, 1995.

[34] D. Sujudi and R. Haimes, “Identification of swirling flow in 3-d vector fields,” in **Proceedings of the 12th Computational Fluid Dynamics Conference**, 1995, pp. 792–799.

[35] R. Peikert and M. Roth, “The ”Parallel Vectors” operator–a vector field visualization primitive,” in **Proceedings of IEEE Visualization ’99**, 1999, pp. 263–532.

[36] D. Bauer and R. Peikert, "Vortex tracking in scale-space," in **Proceedings of the symposium on Data Visualisation 2002**, 2002, pp. 233–ff.

[37] 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 **VIS 05. IEEE Visualization, 2005.** IEEE, 2005, pp. 631–638.

[38] R. Fuchs, "The visible vortex-interactive analysis and extraction of vortices in large time-dependent flow data sets," Ph.D. dissertation, 2008.

[39] S. Kida and H. Miura, "Identification and analysis of vortical structures," **European Journal of Mechanics-B/Fluids**, vol. 17, no. 4, pp. 471–488, 1998.

[40] N. J. Zabusky, O. Boratav, R. Pelz, M. Gao, D. Silver, and S. Cooper, "Emergence of coherent patterns of vortex stretching during reconnection: A scattering paradigm," **Physical Review Letters**, vol. 67, no. 18, p. 2469, 1991.

[41] R. Cucitore, M. Quadrio, and A. Baron, "On the effectiveness and limitations of local criteria for the identification of a vortex," **European Journal of Mechanics-B/Fluids**, vol. 18, no. 2, pp. 261–282, 1999.

[42] J. Kasten, J. Reininghaus, I. Hotz, H.-C. Hege, B. Noack, G. Daviller, and M. Morzynski, "Acceleration feature points of unsteady shear flows," **Archives of Mechanics**, vol. 68, pp. 55–80, 02 2016.

[43] 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**. Center for Turbulence Research, Dec. 1988, pp. 193–208.

[44] J. Jeong and F. Hussain, "On the identification of a vortex," **Journal of Fluid Mechanics**, vol. 285, pp. 69–94, 1995.

[45] A. E. Perry and M. S. Chong, "Topology of flow patterns in vortex motions and turbulence," **Applied Scientific Research**, vol. 53, no. 3, pp. 357–374, 1994.

[46] A. Perry and M. S. Chong, "A description of eddying motions and flow patterns using critical-point concepts," **Annual Review of Fluid Mechanics**, vol. 19, pp. 125–155, 1987.

[47] T. Günther and H. Theisel, "Hyper-objective vortices," **IEEE Transactions on Visualization and Computer Graphics**, vol. 25, no. 1, pp. 1–1, 2018.

[48] H. Theisel, M. Hadwiger, P. Rautek, T. Theußl, and T. Günther, “Vortex criteria can be objectivized by unsteadiness minimization,” **Physics of Fluids**, no. 33, p. 107115, 2021.

[49] G. Haller, “Can vortex criteria be objectivized?” **Journal of Fluid Mechanics**, vol. 508, p. A25, 2021.

[50] P. Petersen, **Riemannian Geometry**, 3rd ed. Springer-Verlag, 2016.

[51] A. McInerney, **First Steps in Differential Geometry**. Springer-Verlag, 2013.

[52] M. Kilian, N. J. Mitra, and H. Pottmann, “Geometric modeling in shape space,” **ACM Transactions on Graphics**, vol. 26, no. 3, p. Article No. 64, 2007.

[53] M. Ben-Chen, A. Butscher, J. Solomon, and L. Guibas, “On discrete Killing vector fields and patterns on surfaces,” in **Proceedings of Eurographics Symposium on Geometry Processing**, 2010, pp. 1701–1711.

[54] J. Martinez Esturo, C. Rössl, and H. Theisel, “Generalized metric energies for continuous shape deformation,” in **Mathematical Methods for Curves and Surfaces**, ser. Lecture Notes in Computer Science. Springer-Verlag, 2012, pp. 135–157.

[55] J. Solomon, M. Ben-Chen, A. Butscher, and L. Guibas, “As-Killing-as-possible vector fields for planar deformation,” in **Proceedings of Eurographics Symposium on Geometry Processing**, 2011, pp. 1543–1552.

[56] O. Azencot, M. Ben-Chen, F. Chazal, and M. Ovsjanikov, “An operator approach to tangent vector field processing,” **Computer Graphics Forum**, vol. 32, no. 5, pp. 73–82, 2013.

[57] O. Azencot, M. Ovsjanikov, F. Chazal, and M. Ben-Chen, “Discrete derivatives of vector fields on surfaces–an operator approach,” **ACM Transactions on Graphics**, vol. 34, no. 3, p. Article No. 29, 2015.

[58] I. Wald, S. Woop, C. Benthin, G. S. Johnson, and M. Ernst, “Embree: A kernel framework for efficient cpu ray tracing,” **ACM Transactions on Graphics**, vol. 33, no. 4, pp. 143:1–143:8, Jul. 2014.

[59] X. Zhang, M. Hadwiger, T. Theul, and P. Rautek, “Interactive exploration of physically-observable objective vortices in unsteady 2d flow,” **IEEE Transactions on Visualization and Computer Graphics**, vol. 28, no. 2, p. to appear, 2022.

[60] 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**, vol. 26, no. 1, pp. 280–290, 2020.

[61] C. Jung, T. Tél, and E. Ziemniak, "Application of scattering chaos to particle transport in a hydrodynamical flow," **Chaos: An Interdisciplinary Journal of Nonlinear Science**, vol. 3, no. 4, pp. 555–568, 1993.

[62] 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**, 2009, pp. 193–204.

APPENDICES

A Lie Algebra

Scalar multiplication: Scalar times Lie algebra element

The scalar multiplication for the vector space structure, e.g., $a\,\vec{e}_1$, where $\vec{e}_1$ denotes a vector field on M, is defined pointwise in each tangent space via standard scalar multiplication. In a word, for any point $x \in M$, the basis vector in that tangent space T_xM, i.e., $\vec{e}_1(x)$, is multiplied by the same coefficient a:

$$(a\,\vec{e}_1)(x) := a\,\vec{e}_1(x). \tag{A.1}$$

Vector addition: Addition of two Lie algebra elements

Likewise, the vector addition for the vector space structure is also defined by pointwise addition of vectors in each tangent space:

$$(\vec{e}_1 + \vec{e}_2)(x) := \vec{e}_1(x) + \vec{e}_2(x). \tag{A.2}$$

Efficient computation of inner products

Instead of computing the above integrals for every arbitrary pair of observers $(\vec{w}_1, \vec{w}_2)$, we can exploit the vector space structure of the Lie algebra of Killing fields. From this structure, we know that every Killing field can be written as a linear combination $\vec{w} = \sum_i w^i\,\vec{e}_i$, where the $\vec{e}_i$ are basis Killing vector fields on the manifold M. Using the inner product $\langle\langle\cdot,\cdot\rangle\rangle$ between two vector fields given by Eq. 4.2, we can define a metric tensor g_{ij} of Killing vector fields, given by the components

$$g_{ij} := \langle\langle\vec{e}_i, \vec{e}_j\rangle\rangle. \tag{A.3}$$

Given this metric tensor, we can then compute the inner product between any two Killing fields simply from their components as

$$\langle\langle\vec{w}_1, \vec{w}_2\rangle\rangle = \sum_{i,j} g_{ij}\, w_1^i w_2^j. \tag{A.4}$$

Linear independence of the Lie algebra basis

The linear independence of our Lie algebra basis needs to be verified as the linear independence of vector fields, not as that for individual vectors in some tangent space. Otherwise, we can think of what will happen in a two-dimensional tangent space: more than two vectors would always be linearly dependent. But we have three linearly independent basis vector fields indeed. We give a definition of the linear independence of three basis fields as following:

Definition 15. *The linear independence of three basis fields $\vec{e}_i, \vec{e}_j, \vec{e}_k$ is given if there are no coefficients $\lambda, \mu \in \mathbb{R}$ such that*

$$\vec{e}_i = \lambda\vec{e}_j + \mu\vec{e}_k. \tag{A.5}$$

for every (cyclic) permutation of $(i, j, k) = (1, 2, 3)$. This equation must be read such that for fixed $\lambda, \mu \in \mathbb{R}$, for all points $x \in M$, the vectors in each tangent space T_xM would have to be

$$\vec{e}_i(x) = \lambda\vec{e}_j(x) + \mu\vec{e}_k(x). \tag{A.6}$$

In our framework, for $M = \mathbb{R}^2$ and the basis given by Eq. 4.3, this is trivial to see for $\vec{e}_i = \vec{e}_3$. However, it is also not hard to see in the other cases. For $M = \mathbb{S}^2$ and the basis given by Eq. 4.4, we can imagine choosing two basis fields to reproduce a non-zero vector of the third basis field at some point, and then considering the "pole" of the third field, where the third field has a critical point (the vector is zero), but the linear combination that we just considered gives a non-zero vector.

The Lie bracket

The Lie bracket is a "vector multiplication" operation defined for a pair of vectors (elements) of a Lie algebra. In our case, these elements are vector fields, and the corresponding Lie bracket is then a map

$$\begin{aligned} [\cdot,\cdot]\colon \mathcal{X}(M) \times \mathcal{X}(M) &\to \mathcal{X}(M), \\ (\vec{u},\vec{v}) &\mapsto [\vec{u},\vec{v}]. \end{aligned} \tag{A.7}$$

Here, $\mathcal{X}(M)$ denotes the space of all possible (smooth) vector fields on the manifold M. This Lie bracket of vector fields is a differential-geometric operator that maps a pair of vector fields on M to another vector field on M. Importantly, if we consider only the Killing vector fields on M, the Lie bracket maps a pair of Killing fields on M to another Killing field on M. The Lie bracket is identical to the Lie derivative of one vector field with respect to the flow of another vector field, i.e.,

$$[\vec{u},\vec{v}] = \mathcal{L}_{\vec{u}}\vec{v}, \tag{A.8}$$

with both sides denoting a vector field. More explicitly, we can compute the vector field $\mathcal{L}_{\vec{u}}\vec{v}$, and thus the Lie bracket $[\vec{u},\vec{v}]$, by

$$[\vec{u},\vec{v}] = \mathcal{L}_{\vec{u}}\vec{v} = \nabla\vec{v}(\vec{u}) - \nabla\vec{u}(\vec{v}), \tag{A.9}$$

where $\nabla\vec{u}$ and $\nabla\vec{v}$ are the velocity gradient tensors of the vector fields $\vec{u}$ and $\vec{v}$, respectively, and we evaluate in the directions $\vec{v}$, $\vec{u}$, respectively.

B Pushforward and Pullback

We denote reference frame transformations by diffeomorphisms ϕ_t, with each map ϕ_t an isometry, from a manifold M to itself. Each map

$$\begin{aligned} \phi_t \colon M &\to M, \\ x &\mapsto \phi_t(x), \end{aligned} \tag{B.1}$$

induces a **differential** (or pushforward) ϕ_t between tangent spaces on M (in fact on the tangent bundle TM). For each tangent space T_xM, i.e., the tangent space at the point $x \in M$, the differential is a linear map

$$\begin{aligned} (\phi_t)_x \colon T_xM &\to T_{\phi_t(x)}M, \\ \vec{x} &\mapsto (\phi_t)_x(\vec{x}). \end{aligned} \tag{B.2}$$

The differential maps vectors from points x to points $\phi_t(x)$.

In Sec. 4.5, for consistency and in order to avoid the frequent use of pushforwards of inverse diffeomorphisms ϕ_t^{-1}, we instead use the **pullback** ϕ_t^*, mapping "back" vectors from points $\phi_t(x)$ to points x by

$$(\phi_t^* \, \vec{x})_x := \left(\phi_t^{-1}\right)_{\phi_t(x)}(\vec{x}). \tag{B.3}$$

For second-order tensors $\mathbf{T}$, linearly mapping vectors to vectors, we in any case require the corresponding pullback ϕ_t^* of $\mathbf{T}$, defined by

$$\begin{aligned} (\phi_t^* \mathbf{T})_x \colon T_xM &\to T_xM, \\ \vec{x} \mapsto (\phi_t^* \mathbf{T})_x(\vec{x}) &:= \phi_t^{-1}\big(\mathbf{T}\big(\phi_t(\vec{x})\big)\big). \end{aligned} \tag{B.4}$$

While pullbacks in general are defined for smooth maps that need not be diffeomorphisms, the above definitions require the map ϕ_t to be a diffeomorphism (guaranteed to have an inverse) to allow mapping back vectors in the inverse direction ϕ_t^{-1}, which in our context is fulfilled.

C Observer Transformation on the Sphere

We now concretely give the reference frame transformation, described in general and for the Euclidean plane in Sec. 4.5.1, for the sphere.

Frame change diffeomorphism (isometry) on the sphere

The time-dependent isometry ϕ_t for frame transformation on the sphere, corresponding to a rigid motion (i.e., rotation) of the sphere, is given by

$$\phi_t(x) = \vec{r}^{-1}\big(\mathbf{R}(t)\ \vec{r}(x)\big). \tag{C.1}$$

Here, $\vec{r}(x)$ is a 3D vector pointing from the center of the sphere to the point x on the sphere, embedded in $\mathbb{R}^3$, $\mathbf{R}(t)$ is the total integrated 3×3 rotation matrix giving the rigid transformation of the sphere at time t, and $\vec{r}^{-1}(\cdot)$ denotes the inverse of $\vec{r}(\cdot)$ interpreted as a function, mapping "back" a vector pointing from the center of the sphere to the point $\phi_t(x)$ embedded in $\mathbb{R}^3$, to the "intrinsic" point $\phi_t(x)$ on the sphere.

Pullback on the sphere

The corresponding pullback ϕ_t^* of a vector field $\vec{x}$ on the sphere is

$$\phi_t^* \, \vec{x} = (\mathbf{B}^*)' \, \mathbf{R}^T(t) \, \mathbf{B} \, \vec{x}. \tag{C.2}$$

The pullback ϕ_t^* of a second-order tensor field $\mathbf{T}$ on the sphere is

$$\phi_t^* \, \mathbf{T} = (\mathbf{B}^*)' \, \mathbf{R}^T(t) \, \mathbf{B} \;\, \mathbf{T} \;\, \mathbf{B}' \, \mathbf{R}(t) \, \mathbf{B}^*. \tag{C.3}$$

The $\mathbf{R}(t)$ are the same rotation matrices as above. The 3×2 matrices $\mathbf{B}$, $\mathbf{B}^*$ map vectors $\vec{x}$, at $\phi_t(x)$, and $\vec{x}^*$, at x, respectively, from two components referred to tangent space bases embedded in $\mathbb{R}^3$, $\{\vec{b}_1, \vec{b}_2\}$, at $\phi_t(x)$, and $\{\vec{b}_1^*, \vec{b}_2^*\}$, at x, respectively, to their embedding in $\mathbb{R}^3$. The corresponding 2×3 matrices $\mathbf{B}'$, $(\mathbf{B}^*)'$ perform the "inverse" operation, mapping 3D vectors, tangent to the sphere, back from three to two components, again referred to the bases $\{\vec{b}_1, \vec{b}_2\}$, $\{\vec{b}_1^*, \vec{b}_2^*\}$, respectively.

In case the vectors are given embedded in $\mathbb{R}^3$, instead of as intrinsic 2D vectors, and the tensors $\mathbf{T}$ are likewise given embedded in 3D, the above matrices $\mathbf{B}$, $\mathbf{B}^*$, $\mathbf{B}'$, $(\mathbf{B}^*)'$ simply become identity matrices.

D Supplementary Materials

The illustration video is available at:

`http://vccvisualization.org/research/killinginteraction/`.

E Papers Submitted and Under Preparation

- Xingdi Zhang, Markus Hadwiger, Thomas Theußl, Peter Rautek, “Interactive Exploration of Physically-Observable Objective Vortices in Unsteady 2D Flow” [59], **IEEE Transactions on Visualization and Computer Graphics (Proceedings IEEE VIS 2021)**, Vol. 28, No. 2, 2022.