Pattern Recognition 45 (2012) 3912–3925, Elsevier, 2012.

Detecting Singular Patterns in 2-D Vector Fields Using Weighted Laurent Polynomial

Wei Liu and Eraldo Ribeiro*

Computer Vision Laboratory
Department of Computer Sciences
Florida Institute of Technology
Melbourne, FL 32901, U.S.A.

Abstract

In this paper, we propose a method for detecting patterns of interest in vector fields. Our method detects patterns in a scale- and rotation-invariant manner. It works by approximating the vector-field data locally using a Laurent polynomial weighted by radial basis functions. The proposed representation is able to model both analytic and non-analytic flow fields. Invariance to scale and rotation is achieved by combining the linearity properties of the model coefficients and a scale-space parameter of the radial basis functions. Promising detection results are obtained on a variety of fluid-flow sequences.

Keywords: vector fields, singular-pattern detection, scale- and rotation-invariance, complex-valued function, Laurent polynomials

1. Introduction

The ability to automatically detect patterns of interest in vector fields is useful in many applications such as texture analysis [1, 2], fingerprint classification [3, 4], and fluid analysis [5, 6, 7]. Some relevant patterns include convergence (divergence) flows such as sources (sinks) and rotational flows such as vortices. Broadly speaking, detecting interest patterns in vector fields is similar to detecting interest features in scalar images [8, 9]. Once detected, these features are compactly represented by descriptors that can subsequently be used for solving higher-level problems such as flow-field recognition [5], visualization [10], classification [3, 4], and tracking [7, 6].

Most vector-field descriptors are limited in two aspects. First, many approaches use symbolic representations [1, 2, 3] that are less sensitive to subtle appearance variations. Secondly, rotation and scaling can be an issue with vector-field descriptors, in contrast to modern image-feature detectors [8, 9].

Our main contribution is an algorithm for detecting and describing patterns in 2-D vector fields. The proposed method compensates for scale and rotation transformations. We commence by modeling vector fields locally using a Laurent polynomial (Section 3), in which the model coefficients provide a compact representation that we use to measure similarity between vector-field regions. The complex-domain monomials used in our approximation model allows us to handle vector-field rotations and scalings *algebraically* (Sections 4 and 5). Following the results reported in [11], we combine these ideas into a SIFT-like multi-scale detection algorithm that searches the vector-field scale-space for singular patterns as local maxima of a pre-defined energy function (Section 6). We evaluated our detector on both synthetic turbulent flow fields and real-world satellite images (Section 8). The method produced very good detection results on flow fields undergoing rotation and scaling, which demonstrates the potential of our method.

2. Related work

In contrast with scalar images that are often unstructured and contain many discontinuities, vector-field data usually originate from continuous physical processes such as fluid motion and natural dynamic textures. Thus, it is expected that model-based approaches for singular-pattern detection in vector fields are common in the literature. For this class of approaches, the choice of model requires special attention regarding the balance between simplicity and descriptiveness. Here, the simplest model is a template flow, making template-matching a common technique of many detection methods based on convolution [12], correlation [13] or filtering operations [4]. While these detectors generally work well, the detected patterns are often required to match the template's size, shape, and orientation. Further descriptive flexibility is added to these methods by calculating the Winding number (or Poincaré index) [14]. This number has been used for locating singular patterns in fingerprint indexing and classification [15, 3]. However, noise sensitivity in its calculation is a common issue.

Some model-based methods use locally affine models for detecting singular points in vector fields. These approaches categorize flow fields into symbolic classes using the linear phase-portrait method [1]. An extension to a nonlinear flow model was proposed by Ford et al. [16], and has been used for fingerprint classification [17]. Recent work by Kihl et al. [18] improved it further to detect multi-scale singular points.

*Corresponding author: eribeiro@cs.fit.edu

Table 1: Related Work

Year	Representative works	Invariance			Application
		Laminar	Scale	Rotation	
1992	Rao and Jain [1]	No	No	Yes	Texture
1993	Sherlock and Monro [19]	No	No	No	Fingerprint
1997	Nogawa et al. [2]	No	No	Yes	Texture
2003	Nilsson and Bigun [4]	No	Yes	Yes	Fingerprint
2003	Gu et al. [20]	No	No	No	Fingerprint
2004	Copetti et al. [5]	Yes	No	Yes	Fluid
2006	Li et al. [17]	No	No	Yes	Fingerprint
2007	Schlemmer et al. [13]	No	Yes	Yes	Visualization
2008	Kihl et al. [18]	No	Yes	Yes	Fluid
2008	Fan et al. [3]	No	No	Yes	Fingerprint
2009	Wong and Yip [21]	No	Yes	Yes	Fluid
2009	Xu et al. [7]	No	No	Yes	Fluid
2010	**Our work [11]**	Yes	Yes	Yes	Fluid

Two-dimensional vector fields can also be represented using complex-domain functions, which provide a more compact representation for vector fields than their real-valued counterparts. For instance, Shelock and Monro [19] used the complex zero-pole model to represent and detect singular points in fingerprint images. Nogawa et al. [2] modeled singular patterns based on Cauchy's residue theorem. Possible issues in this case are noise sensitivity as well as the presence of multiple singular points. Multiple singular points can be addressed by means of a complex Laurent series as shown by Zhou and Gu [22] in their representation of fingerprint's ridge-flow patterns. The method we propose in this paper also uses a linear combination of negative and positive powers of $z \in \mathbb{C}$. However, instead of obtaining a global representation, we focus on local rotation-scale invariant descriptions of patterns in a vector field.

The way singular patterns are defined is key to detection. As in the case of scalar-image features, the definition of singular patterns in vector fields can be ambiguous. As Jiang et al. [23] pointed out, a precise definition does not exist even for basic patterns such as vortices. The term *singular points* is commonly used to describe regions of interest in fluid patterns, even though fluid patterns are characterized by their surrounding flow field. In this paper, we call these regions *singular patterns* (as a region descriptor rather than a point descriptor).

Among all singular patterns, vortex detection is an important, yet difficult problem that has been studied mostly in scientific data visualization [23]. A common approach is to relate vortices to local maxima of flow-field properties such as vorticity [24] and helicity [25]. Another approach is to analyze the characteristics of patterns surrounding singular points. For example, Sadarjoen et al. [26] examines the rotation angle of the surrounding streamlines, and then groups the ones that share common central points. This method not only detects the vortices, but also determines their sizes (scales). Nevertheless, these methods are restricted to vortex detection and provide little descriptiveness for general singular patterns.

A natural way to locate singular patterns is to define their position to be at the vanishing points of vector fields (i.e., the points at which the flow is zero). This definition, common in approaches for directional texture and fluid analysis [1, 13, 18, 21], is simple and includes many interesting patterns such as vortices, sinks, and sources. However, vanishing-point locations may become corrupted by the presence of a background (or laminar) flow. This problem can be minimized by decomposing flow fields into background and foreground flows [5]. The ability to handle the presence of background flows is important to practical applications such as the analysis of weather images; as patterns generated by hurricanes, which may have their centers shifted when rotating as a vortex [5].

When complex-valued functions are used to represent flow fields, singular patterns are often defined as discontinuities, and coincide with the singularities in the differentiability sense [19, 22, 27]. Some complex-valued models are specifically designed to represent vortices and sinks (or sources), e.g., the Rankine models [5]. Despite their compactness, these definitions can represent only a subset of the possible vanishing points. In this paper, we also use complex-valued functions as our representation, but we define singular patterns as local maxima of a multi-scale singular-energy function (Section 6). Our definition includes classic vanishing points. Additionally, our model implicitly handles background flows; making our singular-pattern detection method fairly robust to background flows in a way similar to that of the Galiean invariant defined in [23].

Few approaches achieve scale and rotation invariance in the description of vector-field patterns, presenting a clear contrast with available affine-invariance feature descriptors for scalar images [8, 9]. For some vector-field descriptors, rotation invariance is implicitly achieved by adopting simplistic symbolic descriptions [2, 5]. Rotation invariance can also be achieved from geometric properties such as symmetry [4], but these descriptors can be restricted to specific types of flow patterns such as vortices [2, 5] or to applications such as fingerprint analysis [4]. Recently, Fan et al. [3] detected singular patterns of different orientations based on a Hough-transform voting scheme, and Xu et al. [7] directly extended the spin-image method for scalar images to describe vector fields. Wang et al. [27] studied the rotation of fingerprint singular patterns using analytical models based on the Fourier Transform.

On the issue of achieving scale invariance, researchers have attempted to adapt the scale-space theory of scalar images to vector fields [28]. However, the scale-space concept has not been extensively used in detecting multi-scale flow patterns. For example, Kihl et al. [18] detect singular patterns of different scales by varying the size of searching windows, and Wong et al. [21] fit a scale-parameterized model to flow fields. Multi-scale detection has also been performed through multi-scale filtering [29] or multi-resolution analysis [30]. In our work, we adopt a scale-space representation that allows us to accurately handle different scales of flow patterns.

A recent work by Schlemmer et al. [13] developed the equivalence of Hu moment invariants [31] for flow fields, resulting in a scale-rotation invariant descriptor. Nevertheless, their detection method is a type of template-matching, and is limited by the template's shape. To the best of our knowledge, only Schlemmer et al. [13] and our method provide a framework for detecting and describing general singular patterns in flow fields, while *simultaneously* addressing invariance to both scale and rotation transformations. Recently, Marquering et al. [32] has

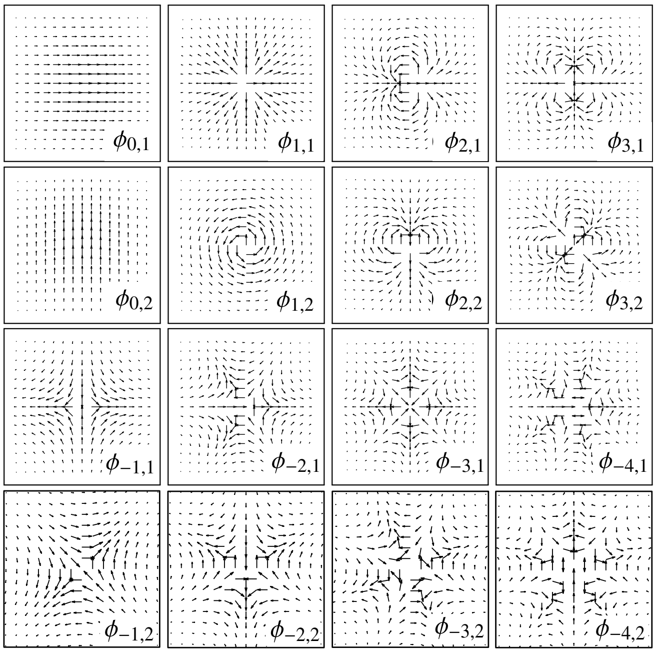


Figure 1: Basis flows $\phi_{k,j}$ for $j = 1, 2$. Row 1: flows derived from z^k for $k = 0, \ldots, 3$. Row 2: flows derived from iz^k for $k = 0, \ldots, 3$. Row 3: flows derived from z^k for $k = -4, \ldots, -1$. Row 4: flows derived from iz^k for $k = -4, \ldots, -1$.

proposed an interesting application of our method [11] for the detection of singular patterns in blood flows.

Finally, complex-valued representations and transformations have also been studied in the context of filtering. For example, Reisert et al. [33] used holomorphic filters for analyzing the orientation of flow fields, and Wang et al. [34] studied the rotational invariants of polar Fourier Transforms in a similar fashion to Zernike moment invariants. Rather than finding numerical invariants, our method aligns flow fields to a predefined set of basis flows to achieve a rotationally invariant comparison. In Table 1, we summarize some of the related work discussed in this section based on their abilities to handle background flow, scaling, and rotation transformations.

3. Modeling vector fields

Our main goal is to detect regions presenting some degree of singularity in 2-D vector fields. We call these regions *singular patterns*. In this section, we describe how we can model vector fields locally using a power series of complex monomials. These complex monomials are weighted by radial basis functions to produce a scale-space representation for multiscale pattern detection. The model coefficients are calculated by projecting the input vector field onto the monomial functions using cross-correlation as an inner-product operator.

3.1. Local approximation of 2-D vector fields as a power series

Broadly speaking, a vector field is defined by a function that assigns a magnitude and direction (i.e., a vector) to each point of its domain. A 2-D vector field can be intuitively represented by a complex-valued function F defined on a finite domain $\Omega \subset \mathbb{C}$. Because our goal in this paper is to detect singular patterns rather than modeling vector fields exactly, we approximate field F locally as a linear combination of simpler vector fields (i.e., basis vector fields) as follows:

$$F(z) \approx f(z) = \sum_{k=-L}^{N} a_k \phi_k(z, \sigma), \tag{1}$$

where $a_k \in \mathbb{C}$ are the coefficients, and $\phi_k(z, \sigma)$ are complex monomials z^k weighted by the following radial basis function:

$$\phi_k(z, \sigma) = \begin{cases} \frac{1}{C_k} \, \|z\|^{-k} \, z^k \, \exp\left(-\|z\|^2/\sigma^2\right) & \text{if } \|z\| > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Here, C_k is a normalization constant to ensure that the squared magnitudes of the vectors in the basis vector fields sum to unity.

The radial basis function in (2) plays two main roles. It limits the size of the local support of z^k, providing some degree of locality control for the approximation [33]. This control is desirable because the monomials z^k are globally supported, and the magnitude of the vector fields represented by them grows fast as a function of their distance to the origin. In addition, we use the radial basis function to establish a scale-space representation for singular-pattern detection (See Section 6.2).

Equation 1 resembles a Laurent polynomial [35, 36], which is a complex power series with negative and positive powers (originally without the radial weighting). If only the nonnegative powers were used (i.e., $L = 0$) then Equation 1 would be similar to a Taylor series, which can represent only vector fields modeled by analytic functions, therefore excluding important fields such as the ones originated from the shear and conjugate flows [11]. By contrast, the Laurent polynomial is more general and is able to represent, under certain conditions, vector fields modeled by non-analytic functions.

The approximation in Equation 1 is related to existing models such as the Rankine (Appendix B.1) and the phase-portrait (Appendix B.2). In addition, our model can be related to homomorphic filters [33] and spherical Fourier transforms [34], by rewriting the complex monomials using polar coordinates.

3.2. Calculating the approximation coefficients

Let us now focus our attention on the problem of estimating the coefficients a_k of the polynomial in (1). These coefficients can be calculated by using an inner product to project a local vector field f onto the basis functions ϕ_k as follows:

$$\begin{aligned} a_k &= \int_{\mathbb{C}} f(z)\, \overline{\phi_k}(z, \sigma)\; dz \\ &= \underbrace{\int_{\mathbb{C}} f(z) \cdot \phi_k(z, \sigma)\, dz}_{\text{real part}} + i \underbrace{\int_{\mathbb{C}} f(z) \cdot i\, \phi_k(z, \sigma)\, dz}_{\text{imaginary part}} \\ &= a_{k,1} + i\, a_{k,2}, \end{aligned} \tag{3}$$

where $\overline{\phi_k}$ is the complex conjugate of ϕ_k, and the $\cdot$ symbol is the dot product between the vectors in the input vector field f

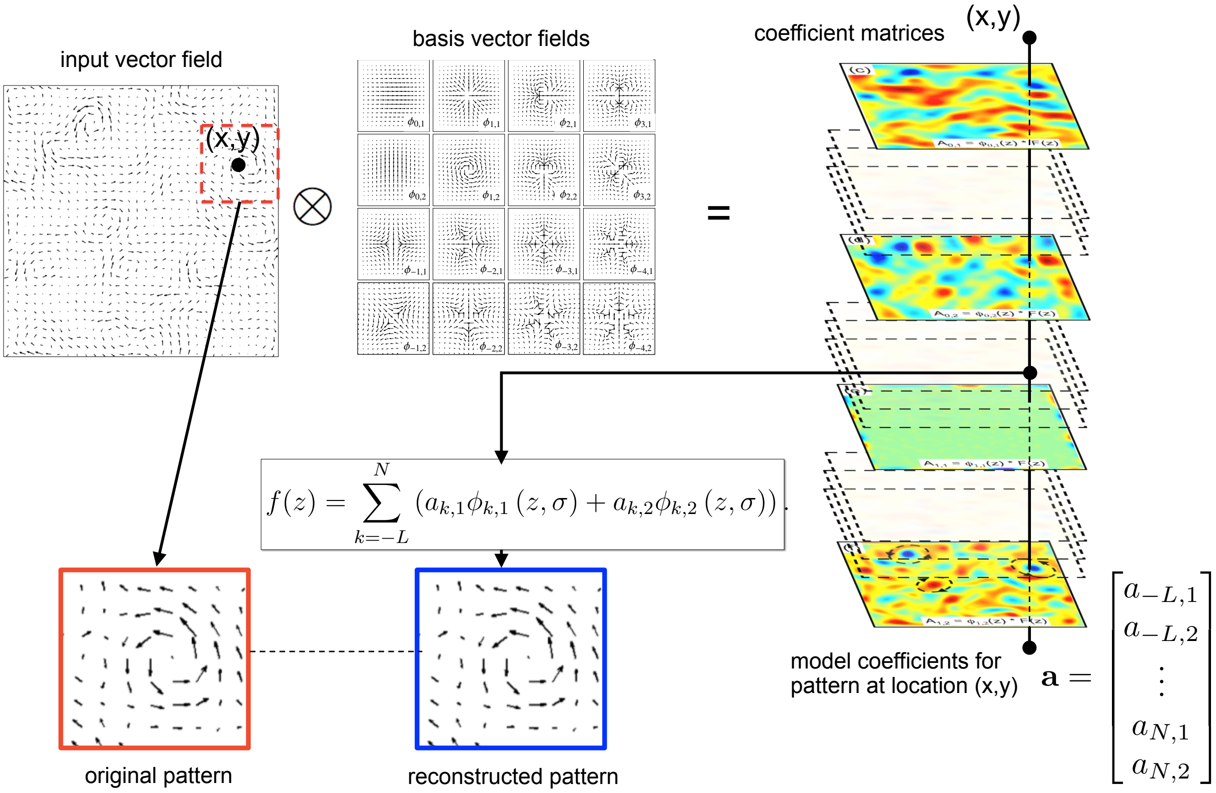


Figure 2: Overview of our vector-field representation. The input vector field is projected on the basis flows using cross-correlation. Vector **a** contains the model coefficients represent the flow appearance at a position (x, y) on the vector field. Basis flows are weighted by radial basis functions that help control the locality of the approximation (scale parameter σ). Vector flow can be reconstructed using the approximation model (Equation 1).

and the vectors in the basis vector field ϕ_k (i.e., if $f(z) = (a, b)^\mathsf{T}$ and $\phi_k(z, \sigma) = (c, d)^\mathsf{T}$, then $f(z) \cdot \phi_k(z, \sigma) = (a, b)(c, d)^\mathsf{T}$). To simplify the notation, we define the following operator to represent the integral calculations in the real and imaginary parts of Equation 3. Given two vector fields f and g, we define:

$$\langle f, g \rangle = \int_{\mathbb{C}} f(z) \cdot g(z)\, dz. \qquad (4)$$

Here, the dot product $\cdot$ can be similarly explained using the standard inner product on vector space.

Let us now briefly focus our attention on Equation 3. It shows that the real and imaginary parts of the Hermitian inner-product (i.e., $a_{k,1}$ and $a_{k,2}$) can be calculated through simple dot-product operations between vectors of the vector field f and of the basis flow ϕ_k. We also note that vector fields generated by the term $i\phi_k$ that appear in the imaginary part of (3) correspond to a counter-clockwise 90-degree rotation of the vectors in ϕ_k. We will denote the monomials ϕ_k and $i\phi_k$ as $\phi_{k,1}$ and $\phi_{k,2}$, respectively, and use them as the basis functions of our approximation model. This extended set of monomials has the advantage of making explicit the modeling of patterns such as vortices (which is given by $i\phi_1$). Given this updated notation, we write the calculation of $a_{k,1}$ and $a_{k,2}$ as:

$$a_{k,j} = \langle f(z), \phi_{k,j}(z, \sigma) \rangle \quad \text{for} \quad j = 1, 2. \qquad (5)$$

The operator $\langle ., . \rangle$ is the vector-field dot product in Equation 4. By using the coefficients $a_{k,1}$ and $a_{k,2}$ and the two basis monomials $\phi_{k,1}(z, \sigma)$ and $\phi_{k,2}(z, \sigma)$, we can rewrite the vector-field approximation in (1) as follows:

$$f(z) = \sum_{k=-L}^{N} \left[a_{k,1}\, \phi_{k,1}(z, \sigma) + a_{k,2}\, \phi_{k,2}(z, \sigma) \right]. \qquad (6)$$

We conclude the description of our vector-field representation by observing that, because we will be calculating the local approximation at every point z_0 of vector field F, the values of the coefficients $a_{k,1}$ and $a_{k,2}$ in (5) can be computed using the cross-correlation (sliding dot-product) between $F(z)$ and $\phi_{k,j}(z, \sigma)$ for $j = 1, 2$, which is given by:

$$\begin{aligned} \langle f(z), \phi_{k,j}(z, \sigma) \rangle &= (F \otimes g)(z_0) \\ &= \int_{\mathbb{C}} F(z + z_0) \cdot \phi_{k,j}(z, \sigma)\, dz. \end{aligned} \qquad (7)$$

Equation 7 results in a matrix of coefficient values for each set of basis flows, i.e., $A_{k,j} = F(z) \otimes \phi_{k,j}(z; \sigma)$, for $j = 1, 2$.

The local vector field at a point z_0 can then be represented by a vector $\mathbf{a} = (a_{-L,1}, a_{-L,2}, \ldots, a_{N,1}, a_{N,2})^\mathsf{T}$ of size $2(N+L+1)$ constructed by concatenating the values of the coefficients in the $A_{k,j}$ maps for each point z_0 (Figure 2). Equation 7 is similar to the correlation operator used in [13], and can be implemented efficiently using the Fast Fourier Transform (FFT).

In the next two sections, we show how rotation and scaling transformations affect our vector-field model.

4. The effect of rotation

The rotation of a vector field f by an angle θ can be defined by the following operator:

$$\Gamma_\theta\left(f(z)\right) = e^{-\theta i} f(z e^{\theta i}). \tag{8}$$

The term $e^{-\theta i}$ is the contravariant factor that ensures the coordinate invariance [37]. In contrast to scalar images, rotation of vector fields not only involves a coordinate transformation, but also a subsequent rotation of the vectors themselves (i.e., a contravariant transformation). Further information about covariant and contravariant transformations of vectors can be found in [37]. Because rotation is a linear transformation, rotating a vector field in our representation corresponds to a sum of rotated basis functions $\phi_k(z)$, i.e., if we apply the operator in (8) to Equation 6 and re-arrange the summation terms, we obtain:

$$\Gamma_\theta\left(f(z)\right) = \sum_{k=-L}^{N} \left(a_{k,1}\Gamma_\theta\left(\phi_{k,1}(z)\right) + a_{k,2}\Gamma_\theta\left(\phi_{k,2}(z)\right)\right). \tag{9}$$

Interestingly, our choice of basis monomials z^k and iz^k are eigenfunctions of the operator Γ_θ because rotating the basis monomials results in scaled versions of the monomials themselves[1], i.e.,

$$\Gamma_\theta(z^k) = e^{(k-1)\theta i}\, z^k \quad \text{and} \quad \Gamma_\theta(iz^k) = e^{(k-1)\theta i}\, iz^k, \tag{10}$$

where $e^{(k-1)\theta i}$ are eigenvalues. By plugging (10) into (9) and rearranging the terms containing the basis monomials, we obtain:

$$\begin{aligned} a'_{k,1}(\theta) &= \cos\left((k-1)\theta\right) a_{k,1} - \sin\left((k-1)\theta\right) a_{k,2} \\ a'_{k,2}(\theta) &= \sin\left((k-1)\theta\right) a_{k,1} + \cos\left((k-1)\theta\right) a_{k,2}, \end{aligned} \tag{11}$$

where $a_{k,j}$ and $a'_{k,j}$ are the approximation coefficients for the original and rotated flow fields, respectively. Thus, the coefficients of the rotated and original vector fields are themselves related by a rotation.

In our method, we want to detect singular patterns regardless of their rotation. Next, we use Equation 11 to devise a simple method to align two vector-field patterns.

4.1. *Principle orientations of a vector field*

There is no simple method to compensate for rotation when comparing two vector fields. A possible solution is to use Equation 11 and exhaustively search for the angle θ that maximizes the mutual alignment between two vector fields. Instead, we propose to find a set of "dominant or principle directions" in a vector field, and then use these "directions" to bring the fields into alignment. For this, we select a set of anisotropic "template flows" out of our basis monomials (i.e., excluding the rotation-invariant vortex and source/sink flows). The monomials $\phi_{k,2}$ are also excluded from the template flows set as they are 90-degree rotations of the vectors in $\phi_{k,1}$. As a result, we choose the subset $\mathcal{T} = \{\phi_{k,1}\}_{k\neq 1}$ to be the template flows set. We then calculate the alignment angle for each pattern $f(z)$ as follows:

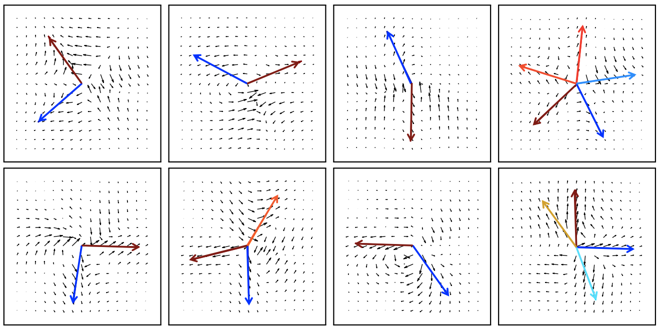


Figure 3: Singular patterns and principle orientations. The colors indicate the relative quality of alignment between singular patterns and the template flows. Brown means the best alignment, and blue indicates the weakest alignment.

$$\begin{aligned} \widetilde{\theta} &= \arg\max_\theta \sum_{k\neq 1} \langle \Gamma_\theta(f(z)), \phi_{k,1} \rangle \\ &= \arg\max_\theta \sum_{k\neq 1} a'_{k,1}(\theta) \\ &= \arg\max_\theta \sum_{k\neq 1} \cos\left((k-1)\theta\right) a_{k,1} - \sin\left((k-1)\theta\right) a_{k,2}, \end{aligned} \tag{12}$$

where $\langle .,. \rangle$ is the vector-field dot product given by Equation 4.

The above maximization can be performed using efficient algorithms (e.g., companion matrix method [38]). The inherent directional nature of vector fields means that (12) may have multiple solutions (at most $2N$ roots [39]). We call these values of θ the *principle orientations* of the vector field, and we use them in our method to bring detected patterns into alignment. In practice, instead of trying to find the "best" single orientation, we accept all the principle orientations for which Equation 12 exceeds a pre-defined threshold. Figure 3 shows examples of vector fields and their corresponding principal orientations as detected by maximizing Equation 12.

5. The effect of scaling

The scaling of vector fields also requires a contravariant transformation to scale vectors following the coordinate transformation. Consider the scaling operator $\Psi_s(.)$, with $s > 0$ given by:

$$\Psi_s(f(z)) = s\, f(s^{-1} z). \tag{13}$$

[1] We dropped the radial basis functions from the formulation as they are rotation invariant.

The scaling operator is also a linear transformation, and its effect on our approximation model is reflected on the basis functions $\phi_{k,j}(z,\sigma)$ as follows (See Appendix A for derivation):

$$\begin{aligned}\Psi_s(\phi_{k,j}(z,\sigma)) &= s\,\phi_{k,j}(s^{-1}z,\sigma)\\ &= s\,\phi_{k,j}(z,s\sigma),\end{aligned} \tag{14}$$

for $j = 1, 2$. Therefore, the scaled version of our basis functions are obtained by scaling the variance of its Gaussian weighting function followed by a multiplication by the scale factor s.

6. Detecting singular patterns in vector fields

In this section, we define a vector-field singularity measure and propose an adapted version of the SIFT algorithm [8] for locating singular patterns in the vector-field scale-space.

6.1. Measuring singularity in vector fields

To detect singular patterns in vector fields we need to define a measure of singularity. In this section, we define a measure of how "singular" a vector field is based on the coefficients a_k. Starting from the vanishing-point definition [18, 1, 28], we know that the flow field vanishes at the center of singular patterns, i.e., $f(z) = 0$ when $z = 0$. If we consider the vector field approximation in Equation 6, then $f(0) = 0$ implies that coefficients $a_{0,1}$ and $a_{0,2}$ are equal to zero. In other words, a local flow pattern around a singular point can be approximated by basis functions $\phi_{k,j}(z,\sigma)$ with $k \neq 0$. We will call the set $\{\phi_{k,j}\}_{k\neq 0}$, $j = 1, 2$, the *singular basis*, and separate the vector field's local expansion into two main components, namely, the constant *background flow* expanded by $\phi_{0,1}$ and $\phi_{0,2}$, and a *singular component* flow expanded by the singular basis.

Decomposing the flow into constant and singular components achieves a similar laminar background separation as in [5]. Nevertheless, the decomposed background flow is only an estimation. As discussed in previous works [5, 18, 1], accurate background-flow estimation is considered a separate problem, often requiring prior knowledge about the flow field. Finally, the constant-flow separation makes our definition of singular patterns rather general compared to singular points defined in [18, 1, 28], because a flow field may not have any vanishing points when a background constant flow or laminar flow exists.

We now define a *singular energy* function E in terms of the squared sum of the coefficients $a_{k,j}$ associated with the singular basis set $\{\phi_{k,j}\}_{k\neq 0}$ for $j = 1, 2$. The singular energy is given by:

$$\begin{aligned}E(z,\sigma) &= \underbrace{\frac{1}{\sigma^2}\sum_{k=-L}^{-1}(a_{k,1}^2 + a_{k,2}^2)}_{E_{\text{shear}}} + \underbrace{\frac{1}{\sigma^2}\sum_{k=1}^{N}(a_{k,1}^2 + a_{k,2}^2)}_{E_{\text{holom}}}\\ &= \frac{1}{\sigma^2}\sum_{k\neq 0,j}\langle F(z), \phi_{k,j}(z;\sigma)\rangle^2,\end{aligned} \tag{15}$$

where $1/\sigma^2$ is a normalization factor so E is independent of the sizes of the basis flows. E has two parts: E_{holom} which is calculated using the coefficients of the holomorphic basis [11], and E_{shear} which is calculated using the negative-indexed basis flows. E_{shear} is related to flow fields of shearing patterns.

The singular energy E in (15) is a function of both the spatial coordinates (i.e., $z \in \mathbb{C}$) and the scale of the basis flows (i.e., $\sigma \in \mathbb{R}^+$). Local maxima of E are the locations and approximate scales of singular patterns in vector fields. Next, we describe an efficient algorithm to detect these locations by searching for local maxima of E over the spatial and scale dimensions.

6.2. Multi-scale search for singular patterns

We perform multi-scale search for singular patterns using an adapted version of the SIFT detector [8] for scalar images. We search the scale-space σ for local maxima of the function E over a sample of discrete points $\sigma_0 2^{o+t/T}$, where $o \in \mathbb{Z}$ is the *octave index*, $t = 0, 1, \ldots, T-1$ is the *finer-scale index*, and σ_0 is the minimum scale of detected patterns (i.e., *nominal scale*).

However, direct calculation of $E(z,\sigma)$ for large scales is computationally expensive because it involves performing cross-correlations with larger basis flows (i.e., $\phi_{k,j}(z,\sigma)$ with larger support as controlled by σ). Fortunately, the calculation of $E(z,\sigma)$ for large scales can be equivalently achieved by down-sampling the original vector field $f(z)$. From the scaling property of the basis flows (Equation 14), we have:

$$\begin{aligned}E(z; s\sigma) &= \frac{1}{s^2\sigma^2}\sum_{k\neq 0,j=1,2.}\langle F(z), \phi_{k,j}(z; s\sigma)\rangle^2\\ &= \frac{1}{s^2\sigma^2}\sum_{k\neq 0,j=1,2.}\langle F(z), s^{-1}\Psi_s(\phi_{k,j}(z;\sigma))\rangle^2\\ &= \frac{1}{s^2\sigma^2}\sum_{k\neq 0,j=1,2.}\left(\int F(z)\phi_{k,j}(s^{-1}z;\sigma)dz\right)^2\\ &= \frac{1}{s^2\sigma^2}\sum_{k\neq 0,j=1,2.}\left(s\int F(sz)\phi_{k,j}(z;\sigma)\,dz\right)^2\\ &= \frac{1}{s^2\sigma^2}\sum_{k\neq 0,j=1,2.}\left(s^2\int \Psi_{s^{-1}}(F(z))\phi_{k,j}(z;\sigma)\,dz\right)^2\\ &= \frac{s^2}{\sigma^2}\sum_{k\neq 0,j=1,2.}\left(\langle\Psi_{s^{-1}}(F(z)), \phi_{k,j}(z;\sigma)\rangle\right)^2.\end{aligned} \tag{16}$$

Therefore, when calculating E for large values of σ, we can scale down $F(z)$, and use basis flows with smaller support to reduce the computational cost.

In theory, we can choose basis flows with constant scales, and down-sample $F(z)$ to calculate $f(z;\sigma)$. However, down-sampling the original vector-field data should be limited as it introduces numerical errors. Instead, we adopt a hybrid approach: the vector-field data is down-sampled at large-scale intervals called octaves, while within each octave, we vary the basis flows' scale in smaller steps. From Equation 16, the sampled singular energy $E(z;\sigma_0 2^{o+t/T})$ is then calculated as:

$$E(z;\sigma_0 2^{o+t/T}) = \xi \sum_{k\neq 0,j=1,2.}\langle\Psi_{2^{-o}}(F(z)), \phi_{k,j}(z;\sigma_0 2^{t/T})\rangle^2, \tag{17}$$

with $\xi = \frac{2^{2(o-t/T)}}{\sigma_0^2}$. Note that the down-sampling of $F(z)$ can be done recursively because $\Psi_{2^{-o}}(F(z)) = \Psi_{2^{-1}}(\Psi_{2^{-o+1}}(F(z)))$. In our algorithm, we first create a pyramid of flow fields $\Psi_{2^{-o}}(F(z))$ through recursive down-sampling by half, and then generate basis flows of different scales, $\phi_{k,j}(z;\sigma_0 2^{t/T})$. Using both the down-sampled flow fields and the multi-scale basis flows, we calculate the multi-scale singular-energy function $E(z; s\sigma)$, and then search each octave of $E(z; s\sigma)$ for the local maxima. Figure 4 illustrates the singular-energy function calculation, and Algorithm 1 summarizes the detection process.

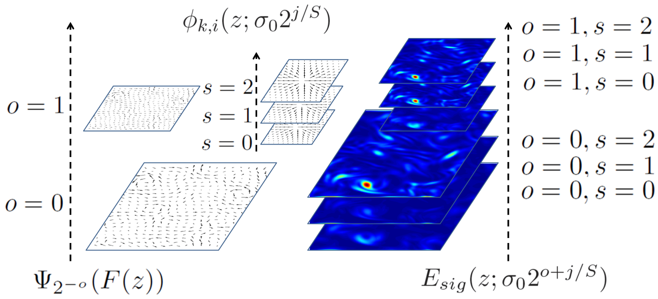


Figure 4: Multi-scale singular-energy function. Left: flow field $F(z)$ is recursively sub-sampled by half into octaves. Middle: a fine-scale pyramid of basis flows is created, completing gaps between each octave. Right: the multi-scale singular energy function $E(z;\sigma_0 2^{o+t/T})$. Octaves are obtained from the cross-correlation between down-sampled flow-field and the pyramid of basis flows.

We further improve the algorithm as follows. First, to capture all local maxima, we overlap octaves by extending the range of index t to $t = -1, 0, \ldots, T$, instead of $t = 0, \ldots, T-1$. Secondly, we refine the positions of the detected local maxima to sub-pixel accuracy through quadratic fitting. These techniques were originally used in the SIFT detector [8].

The time and memory complexity of Algorithm 1 depend on a number of factors, including the sizes of the input vector field and the basis flows, the order (N and L) of the approximation model, the number of octaves N_o and octave sub-sampling factor T. The influence of the size of the input vector field and the size of the basis flows on the computational complexity is similar to the filtering operations in [33]. When compared to the filtering-based method in [33], Algorithm 1 has a higher computational complexity as the subsampling of a scale-space octave and the application of multiple basis flows involve multiple rounds of cross-correlation calculations. It can be inferred from Algorithm 1 that both the time and memory complexity are proportional to $2(N + L + 1)S \log_2 N_o$. For the experiments reported in Section 8, we ran our algorithm on a laptop computer with 4GB of memory and 2.1 GHz dual-core CPU. On average, it took about 4 seconds to process a 256 × 256 input vector field, using 11×11 basis flows of order $L = N = 4$, while scale parameters were chosen as $O = 4$ and $T = 5$. The speed can be further improved using GPU computing [40].

7. Measuring similarity between patterns in vector fields

Before we report on our experimental results, let us define a simple yet effective similarity measure that we use to compare vector-field patterns while compensating for scale and rotation

Algorithm 1: Multi-Scale Singular-Pattern Detection

```
1 Given an input flow $F(z)$, create octaves of $\Psi_{2^{-o}}(F(z))$,
  $o = 0, 1, \ldots, N$ through recursive down-sampling by half, i.e.,
  $\Psi_{2^{-o}}(F(z)) = \Psi_{2^{-1}}(\Psi_{2^{-o+1}}(F(z)))$.
2 Create multiscale bases $\phi_{k,j}(z;\sigma_0 2^{t/T})$, $t = -1, 0, \ldots, T$, and
  $k = 0, 1$. As is SIFT [8], we generate $T + 3$ images to cover a
  complete octave, with nominal scale $\sigma_0 = 1.6$.
3 Calculate the coefficients in each octave using cross-correlation,
  i.e., $a_{k,1}(z;\sigma_0 2^{o+\frac{t}{T}}) = \Psi_{2^{-o}}(F(z)) \otimes \phi_{k,j}(z;\sigma_0 2^{t/T})$
  and
  $a_{k,2}(z;\sigma_0 2^{o+\frac{t}{T}}) = \Psi_{2^{-o}}(F(z)) \otimes \phi_{k,j}(z;\sigma_0 2^{t/T})$,
  for $j = 0, 1$.
4 Calculate the singular energy $E(z,\sigma)$ at each octave.
  $E(z;\sigma_0 2^{o+\frac{t}{T}}) =$
  $\frac{2^{2(o-t/T)}}{\sigma_0^2} \sum_{k\neq 0} \left(\|a_{k,1}(z;\sigma_0 2^{o+\frac{t}{T}})\|^2 + \|a_{k,2}(z;\sigma_0 2^{o+\frac{t}{T}})\|^2\right)$.
5 Detect the singular points at spatial position $(x, y)$ and scale $\sigma$ that
  maximize the singular energy $E(z;\sigma)$.
6 Calculate descriptor and principle orientations (Equation 12) at
  detected positions.
```

transformations. Let f and g be two local vector-field patterns detected using our algorithm (Section 6), with their corresponding scales σ_f and σ_g. The two patterns can be represented by coefficient vectors $\mathbf{a}_f$ and $\mathbf{a}_g$, respectively (Equation 7). If the sets $\{\theta_i\}$ and $\{\theta_j\}$ are the principle orientations of f and g, respectively, obtained from the maximization in (12), then we can define the following similarity measure:

$$d(f, g) = \min_{i,j} \left\| \frac{\mathbf{a}'_f(\theta_i)}{\sigma_f} - \frac{\mathbf{a}'_g(\theta_j)}{\sigma_g} \right\|. \tag{18}$$

Here, we align the patterns to their principle orientations (combinatorial), and take the minimum as their mutual distance. Rotations are done by efficient algebraic calculation (Equation 9).

Next, we use this measure to cluster detected patterns of similar appearance as well as to evaluate the sensitivity of our detector to various degrees of approximation of our model.

8. Experiments

In this section, we report on results obtained using our detection method on both synthetic and real-world flow-field data. On synthetic images, we analyzed the detector's sensitivity with respect to flow-field rotation, scaling, background flows, and random noise. We also evaluated the influence of algorithm parameters on detection repeatability. Flow-field data is also estimated from satellite image sequences, and we use it to demonstrate extraction of patterns in weather systems. To promote reproducibility and for further evaluation of our method, source code and the test data used in our experiments are available online[2].

[2] http://www.cs.fit.edu/~eribeiro/flowdetector/

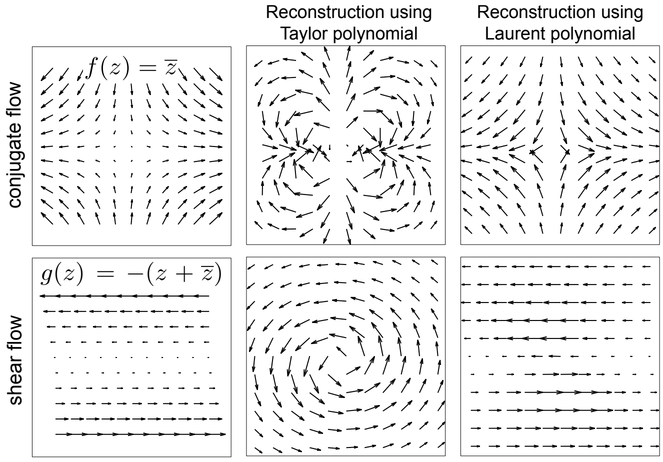


Figure 5: Approximating non-analytic flows. Left: conjugate and shear flows. Middle: approximations using the Taylor polynomial method [11]. Right: approximations using our Laurent polynomial method.

8.1. On representing non-analytic flows

We begin by demonstrating that the representation in Equation 1 provides a good approximation for vector fields generated by non-analytic flows (e.g., conjugate and shear flows).

Figure 5 shows approximation results for two non-analytic flows using the Taylor polynomial method in [11], which uses only nonnegative powers of z, and for the method proposed in this paper (Equation 1). Two non-analytic flow fields are shown: the conjugate flow and the shear flow [41]. Clearly, a linear combination of complex monomials z^k with $k \geq 0$ cannot represent the two flow fields (Taylor polynomial). For the conjugate flow, its analytic approximation does not resemble the original flow. For the shear flow, the Taylor-polynomial captures its analytic component z, but ignores the conjugate part $\bar{z}$. By contrast, our Laurent polynomial gives a (qualitative) better approximation of the conjugate flow by using basis $\phi_{-1,1}$, which consists of the z^{-1} power that is analytic everywhere on $\mathbb{C}$ except at $z = 0$ [42], and its vector field closely resembles that of the conjugate flow (See field $\phi_{-1,1}$ in Figure 1). The shear flow is basically $f(z) = -(z + \bar{z})$ that can be represented by a mixture of the conjugate, $\phi_{-1,1}$, and the source, $\phi_{1,1}$.

By using both negative and positive powers of z as basis monomials, our representation becomes similar to a Laurent series, which is able to model flow fields that cannot be modeled if only positive powers of z were used (e.g., Taylor series). Nevertheless, a single Laurent polynomial cannot represent flows containing multiple singularities. This is not an issue in this paper as our goal is to model individual singular patterns. These multi-singularity flows can be approximated by orthogonal rational functions [3, 42], but it remains unclear how these functions can be used to obtain scale-rotation invariance.

8.2. Synthetic flow fields

We first tested our detector on sequences of turbulent fluids from the European FLUID database [43], and from JHU Turbulence dataset [44]. FLUID contains 2-D turbulent flows of size 256 × 256 which we used to evaluate our detector's repeatability and sensitivity to rotation and scaling transformations. The JHU Turbulence contains 3-D flow-field data sampled on a 1000×1000×1000 cube, thus having a much larger data volume than FLUID. For this dataset, we extracted 2-D slices from it, and used these 2-D flow data to detect large numbers of singular patterns with the goal of evaluating our descriptor on the task of clustering groups of meaningful detected singular patterns.

8.2.1. Detection on FLUID sequences.

Singular patterns detected by our method on a FLUID frame are shown in Figure 6. In the figure, each pattern is enclosed by a circle with radius proportional to the scale of the pattern. Circles are color-mapped according to the value of the pattern's singular energy. This dataset contains sourceless vector fields, and most singular patterns resemble vortices appearing at multiple scales. Our method detected all vortices. Elongated-shaped vortices were detected in pairs. In these cases, some detections could have been discarded based on their singular energy.

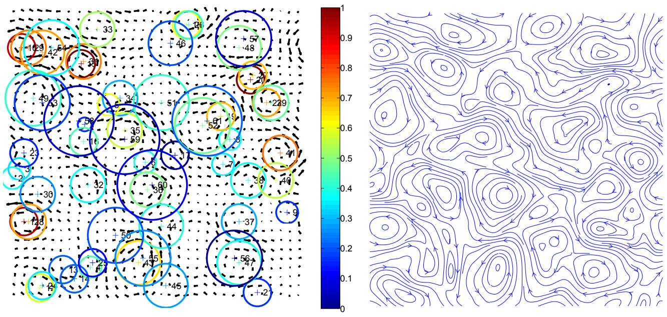


Figure 6: Detection on FLUID dataset. Left: detected patterns. Color indicates relative log-magnitude of singular energy. Vortices are the strongest patterns; Right: flow-field streamlines.

8.2.2. Sensitivity analysis

Using the FLUID database, we first evaluated the impact of the approximation orders N and L on the number of the detected patterns. Table 2 shows the average number of detected patterns in each frame of FLUID using different approximation orders. As the order increased, fewer singular patterns were detected. This effect can be seen in Figure 7, where we show the singular patterns that were detected on the same flow field using the energy functions E as well as its two holomorphic and shearing components as in Equation 15. It is clear that E_{holom} mostly detects vortices, while E_{shear} captures shearing patterns. By combining these two energy functions, E does not necessarily produce more detections. Maybe this happened because higher-order models approximated the continuous flow field better, and thus produced a smoother singular-energy function, leading to fewer local maxima to be detected. The average number of singular patterns per frame converged to about 86.

Then, we evaluated the sensitivity of our detector by matching the singular patterns extracted from flow-fields undergoing rotation and scaling transformations. As it was difficult to evaluate all the possible combinations of flow field transformations,

Table 2: The effect of approximation order on the number of detections. The average number of detected patterns on FLUID decreases with both the maximum positive and negative orders.

	$L = 0$	$L = 1$	$L = 2$	$L = 3$	$L = 4$	$L = 5$	$L = 6$
$N = 1$	150.83	132.11	119.44	117.39	114.83	121.06	121.00
$N = 2$	141.78	115.22	104.00	102.00	100.17	105.71	105.47
$N = 3$	137.17	110.50	97.89	96.78	95.00	99.94	99.70
$N = 4$	128.00	104.06	91.72	91.28	88.83	93.41	93.18
$N = 5$	135.12	108.24	95.88	95.18	92.82	87.11	86.89
$N = 6$	134.76	106.94	94.65	94.06	92.53	86.72	86.56

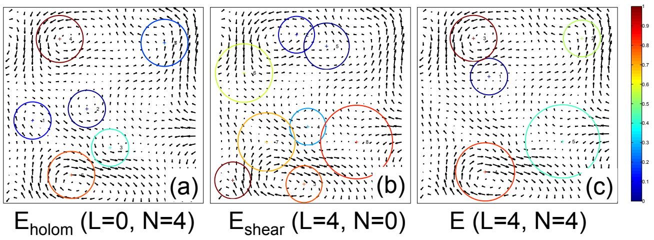


Figure 7: Detected patterns for various singular energies. a) Holomorphic patterns. b) Shearing patterns (these are not captured by E_{holom}). c) Patterns using combined energy (E).

we analyzed the influence of each transformation separately, while keeping the other ones constant. In the first test, we rotated the flow-fields by different angles, but kept the scale unchanged, and matched the singular patterns using the similarity measure given by Equation 18. Figure 8 shows the average accuracy of the matching as a function of the rotation angle. In the second test, we first rotated the flow fields by 45 degrees, and then varied the scale prior to matching the singular patterns using the similarity measure. Figure 9 shows the accuracy as a function of the scaling factor. Because the scale correction might introduce more mismatches between singular patterns of different sizes, the accuracy was slightly lower than in the first test. However, the accuracy was still fairly stable except for the cases when the scale was too small. For these cases, the data loss basically destroyed singular patterns. In both tests, we tried different approximation orders. In general, higher-order models produced better matching accuracy.

In addition to rotation and scaling, we tested the sensitivity of our method for the presence of background flows by adding constant background flows of random directions to the original flow fields. Here, we removed the coefficients related to the background flow from the model before matching the detected singular patterns by simply setting them to zero (i.e., $a_{0,1} = a_{0,2} = 0$). We worked under the assumption that the FLUID dataset was free from background flows. Figure 10(a) shows the matching accuracy as a function of the background-flow magnitude. It is worth pointing out that the maximum magnitude of the original flow fields was only 0.5, allowing them to be completely skewed by the added background flows. With increasing background flow, the matching accuracy dropped significantly for higher-order models due to their numerical sensitivity. However, we believe this sensitivity can be reduced by using a larger nominal-scale parameter σ_0 as the method might be able to estimate background flows more accurately, which in turn could produce less-sensitive results. Figure 10(b) and Figure 10(c) show results produced using the same tests with increasingly larger nominal scales. These results show that the use of higher-order models helps improve detection.

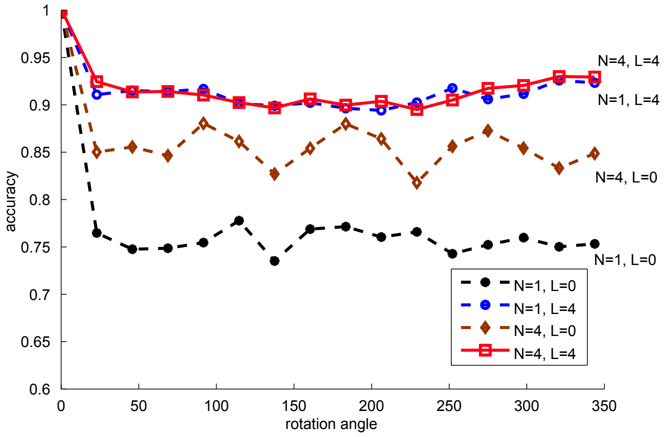


Figure 8: Pattern-detection accuracy as a function of the rotation angle for different approximation orders N and L.

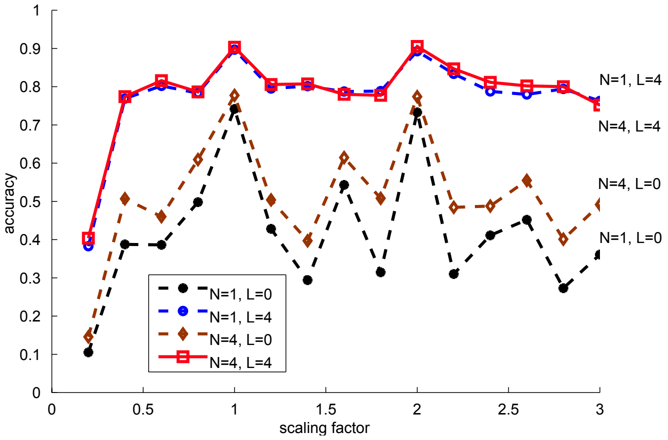


Figure 9: Pattern-detection accuracy as a function of the scaling factor for different approximation orders N and L.

In all pattern-matching experiments, we managed to achieve accuracy rates above 70 percent, and for rotation-scale transformations, more than 80 percent of the singular patterns were correctly matched. In real-world applications, prior, positional, and global information about the singular patterns may allow us to achieve even better matching results.

Finally, we evaluated the sensitivity of our method against the presence of noise in the flow fields. To simulate different noise levels, we added Gaussian random noise of increasing variance v_σ to the FLUID dataset, and ran our detector on the synthetic dataset. We then measured the percentage of singular patterns that were detected and matched to the ones extracted from the original clean flow fields. Here, two singular patterns were matched if they were the closest pair both in

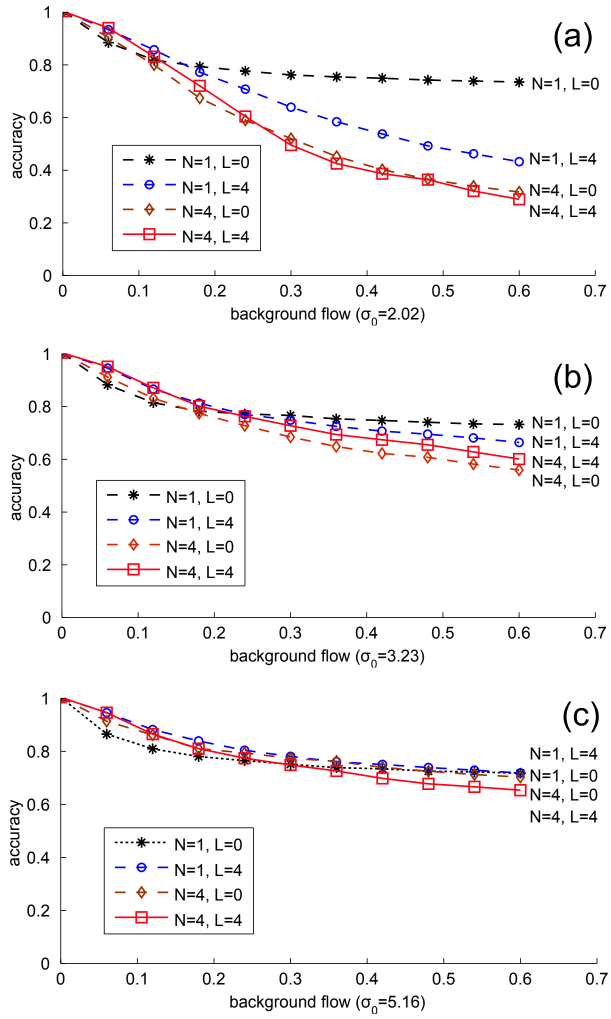


Figure 10: Pattern-detection accuracy against background flow for different nominal scales. Increasing the nominal scale σ_0 increases the robustness of higher-order models. (a) $\sigma_0 = 2.02$, (b) $\sigma_0 = 3.23$, (c) $\sigma_0 = 5.16$.

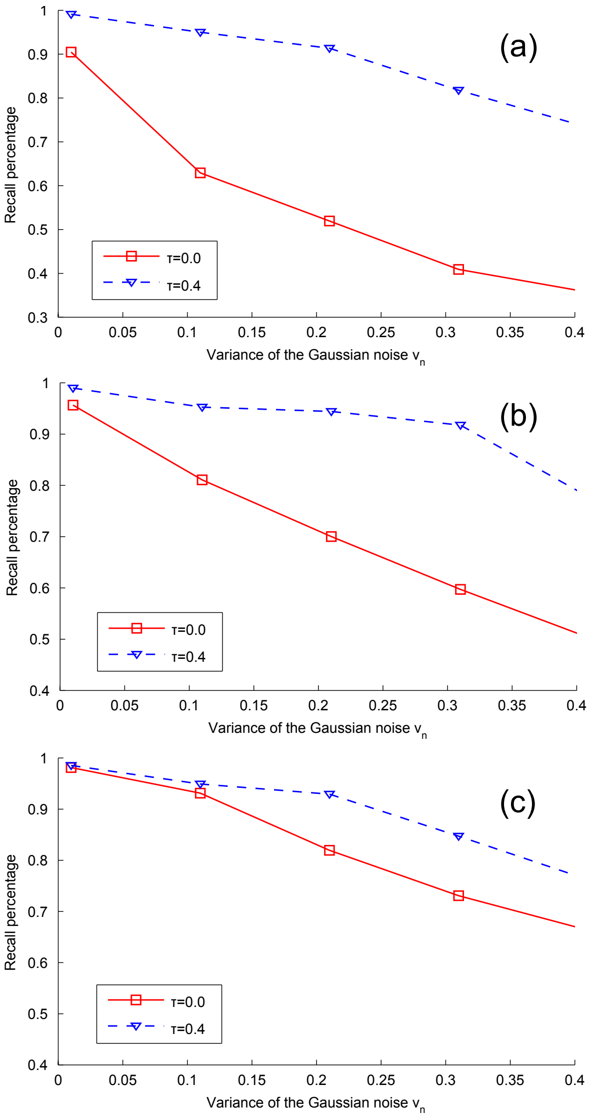
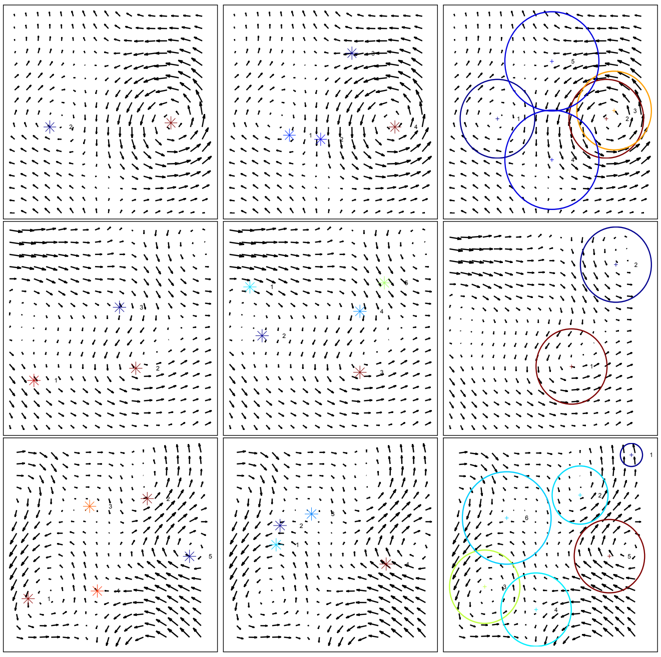


Figure 11: Sensitivity of pattern detection against random noise. Different thresholds τ for singular energy were used to extract patterns. Evaluation was performed for various orders of approximation, including (a) $N = 1$ and $L = 0$, (b) $N = 4$ and $L = 0$, (c) $N = 4$ and $L = 4$.

their approximation coefficients and in their spatial distances. Figure 11 shows the matching percentage as a function of the noise variance v_σ. In addition, we analyzed the influence of three parameters, namely, threshold of the singular energy τ, and approximation orders L and N of our Laurent polynomial model. Singular patterns of high singular energy were more robust to noise. Meanwhile, higher-order approximation seems to help improve the robustness of low-energy singular patterns, but the improvement was not significant for the high-energy ones. When $v_\sigma = 0.41$, the signal to noise ratio (SNR) of the FLUID dataset is already very low (*32* percent). Even at this low SNR, over 70 percent of the singular patterns were still correctly detected by our detector. We believe the robustness of our method is partially due to the use of the cross-correlation operator.

8.2.3. On the phase-portrait and vorticity-based methods

It is difficult to quantitatively compare existing flow-field feature-detection algorithms because of the ambiguity in the definition of flow-field features [23] and the lack of ground-truth data with manually labeled features. To provide a comparison, we implemented versions of two well-known flow-field feature-detection algorithms, namely, the vorticity-based method in [24] and the linear-portrait method [1]. The vorticity-based method simply locates the local maxima of vorticity. For the linear-portrait method, we implemented a singular-point detector (without classifying the types of patterns) using an 11-pixel window size as proposed in [1]. Our goal is to relate the algorithms rather than providing a ranking mechanism. Figure 12 shows detection results for the compared methods. While all three methods produced different results, they mostly agreed on high-energy singular patterns. Compared to our method, the vorticity detector did not detect shearing patterns well. Despite producing many detections, the linear-portrait method was sen-



(a) Vorticity-based (b) Linear-portrait (c) Our method

Figure 12: Comparison of flow-field feature detectors.

sitive to the presence of complicated flow patterns. For this method, incremental changes to the window-size parameter significantly changed its detection results. In addition, only our method estimated the scales of detected patterns.

8.2.4. *Detecting and clustering singular patterns*

In these experiments we used our singular-flow representation for clustering singular patterns detected on the JHU 3-D Turbulence dataset. Here, we selected 2-D slices that were perpendicular to the flow's convecting direction. For better visualization, we separated the detected singular patterns into two groups according to their similarity to vortices and sources (or sinks). This was achieved by examining whether the singular energy was concentrated on the basis functions $\phi_{1,1}, \phi_{1,2}$. If $\|a_{1,1}\|^2 + \|a_{2,2}\|^2$ consisted of more than 50% of the total singular energy, then we labeled the singular pattern as *symmetric*, otherwise, we called it *asymmetric*. Pattern similarity was calculated using the measure in Equation 18.

Detected flow patterns were re-scaled and aligned. For patterns with multiple principle orientations, we generated multiple aligned copies. Each group was clustered into eight clusters using the k-means algorithm. The largest five clusters for symmetric patterns are shown in Figure 13, while the corresponding asymmetric ones are shown in Figure 14. Detected and clustered flows are shown using color-coded directions. Symmetric patterns mostly corresponded to vortices and swirls (i.e., combination of sources and vortices). The flow field we used in this experiment was mostly of a divergent nature. As a result, few sinks were detected, and no sink clusters were obtained. Asymmetric patterns found mostly corresponded to vortices skewed by a background laminar. For better visualization, we further decomposed the asymmetric cluster centers into background and foreground flows. After removing the background flows, we observed that many affine flow fields included conjugate and shearing patterns. Furthermore, most patterns in this group did not have a central vanishing point, yet they still exhibited interesting sudden flow changes. This further indicates the generality of our singular-pattern definition.

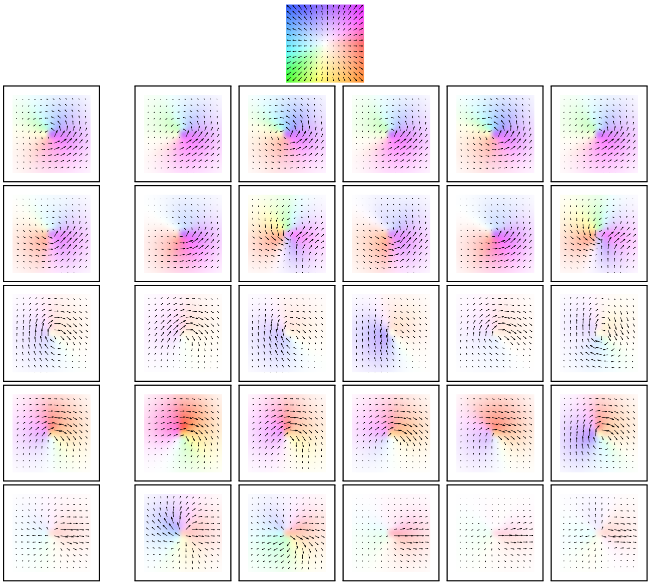


Figure 13: Clusters of symmetric singular patterns. Each row shows a different cluster. Colors indicate vectors' orientation and magnitude according to the color-code legend (top). Left-most column: cluster centers. Columns to the right: samples from the cluster. The five clusters are mostly vortices (counter-clockwise in the first two rows and clockwise in the third and fourth rows). Sources with slight rotation are also present (the fifth row).

8.3. *Real-world flow fields*

We tested our algorithm on satellite images of weather systems from the European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)[3]. The images were taken between March 10th and March 15th, 2010, by the MSG-15 satellite, using the visible-channel sensor VIS-6.

Our method detected interesting patterns in moving clouds including vortices, the sudden appearance and disappearance of clouds, and the shearing motions between neighboring local weather systems. In all experiments we set $L = N = 4$. Cloud motion was estimated from the satellite images using a meshless fluid-motion estimation algorithm [45]. This method produces flow fields that are not biased to lower-order fluid motions, and was considered suitable for the purpose of verifying our singular-pattern detection method.

In Figure 15, we show examples of the satellite images (Column 1) and detected singular patterns (Column 3). For better visualization, we also display the flow fields as stream-lines

[3] http://www.eumetsat.int

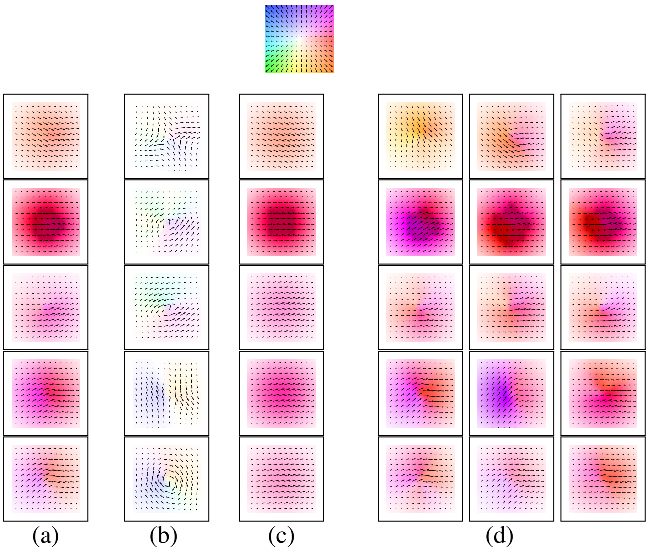


Figure 14: Clusters of asymmetric singular patterns. The first five clusters are shown. Each row shows a different cluster. Colors indicate vectors' orientation and magnitude according to the color-code legend (top). The columns show: (a) cluster centers, (b) cluster centers without background flow, (c) background flows of cluster centers, (d) three instances from the cluster. Without background flows, cluster centers resemble: (1) conjugate flow; (2) counterclockwise vortex skewed by a conjugate flow; 3) counterclockwise vortex; (4) clockwise vortex skewed by a conjugate flow; and (5) clockwise vortex.

with background flows removed (Column 2). Our method detected patterns of different types despite spherical distortions exhibited by the images. Large vortices were detected in all three examples as well as shearing patterns at the boundaries of vortices and neighboring weather systems.

9. Conclusions

We proposed a method for detecting singular patterns in flow fields. Our method approximates flow fields locally using a linear combination of complex-valued basis functions. The new representation can model both analytic and non-analytic flows, as well as affine singular patterns. Classic flow-field models were shown to be related to special cases of our linear-combination model. The detector was tested on many synthetic and real fluid flows by detecting and clustering singular patterns of various rotations and scales.

Limitations of our method include the inability to handle perspective transformations, the higher computational cost and memory need in comparison to related methods, and the restriction to 2-D flow fields only as complex-valued functions cannot represent 3-D flow fields.

Future directions of investigation include the use of geometric algebra (Clifford algebra) [12] to extend our detector to work with 3-D flow fields. In particular, one can try to derive a set of 3-D basis flows using Clifford algebra, and also define principle orientations by aligning a 3-D flow field to a set of pre-defined basis flows. In this case, 3-D multi-scale singular-pattern detection could be achieved by adapting Algorithm 1.

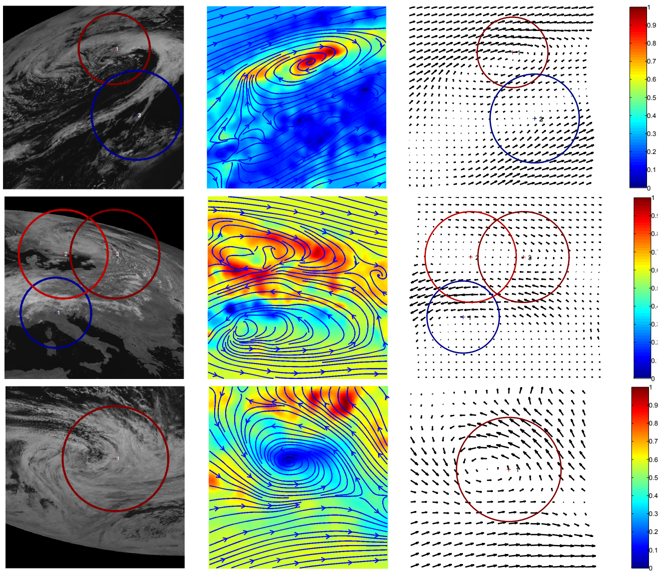


Figure 15: Singular patterns detected on EUMETSAT satellite images. Columns from left to right are: the original satellite images, estimated flow fields with background flow removed, and detected singular patterns.

We plan to explore applications of our detector to different problems such as fingerprint [3, 4], flow-field visualization [10], and biomedical image analysis [32]. Our singular-pattern detector may also be integrated into a fluid-motion estimation method to achieve joint estimation and segmentation. Finally, a real-time implementation of the algorithm using Graphics Processing Units (GPUs) is also desirable.

Appendix A. Scaling transformation

By substituting the basis vector field in Equation 2 into the scaling operator in (13), we have:

$$\begin{aligned}\Psi_s(\phi_k(z,\sigma)) &= s\,\phi_k(s^{-1}z,\sigma)\\ &= s\,\|s^{-1}z\|^{-k}\exp\left(-\frac{\|s^{-1}z\|^2}{\sigma^2}\right)(s^{-1}z)^k\\ &= s\,\|z\|^{-k}\exp\left(-\frac{\|z\|^2}{(s\sigma)^2}\right)z^k\\ &= s\,\phi_k(z,s\sigma). \qquad \text{(A.1)}\end{aligned}$$

We removed the normalization factor $\frac{1}{C_k}$ for simplicity.

Appendix B. Relation to Rankine model and phase portrait

We show the connection between our model with the Rankine model [5]) and the linear phase portrait [18, 1]).

Appendix B.1. The Rankine model

The Rankine model approximates a vortex as a vector field of constant curl inside a disk. Beyond this circular domain the velocity decreases as the inverse squared distance to the disk's center, and the vorticity is null. Locally, the complex representation of this velocity field is given by [5]:

$$f_l(z) = \begin{cases} g_l(z) = -\frac{i\beta_l z}{\|z\|^2}, & \|z\| \geq r_l \\ h_l(z) = -\frac{i\beta_l z}{r_l^2}, & \|z\| < r_l, \end{cases} \tag{B.1}$$

where r_l is the singularity radius, and β_l is its strength. The Rankine model is limited to vortex representation, and has been extended in [5] to represent sink/source patterns as follows:

$$f_l(z) = \begin{cases} g_l(z) = \frac{\alpha_l z}{\|z\|^2}, & \|z\| \geq r_l \\ h_l(z) = \frac{\alpha_l z}{r_l^2}, & \|z\| < r_l. \end{cases} \tag{B.2}$$

Here, α_l represents the local strength, and the *extended* Rankine model represents a source when $\alpha_l > 0$, and a sink when $\alpha_l < 0$. In [5], these two models are linearly combined to form other flow fields. These models are based on holomorphic functions and these can be well approximated by our holomorphic basis flows $\{\phi_{k,j}\}_{k\geq 0,j}$. However, our model can also represent singular patterns that cannot be represented by the Rankine models, such as the conjugate and shearing flows.

Appendix B.2. Phase portrait models

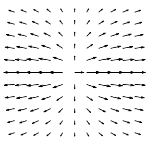
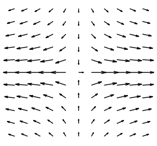
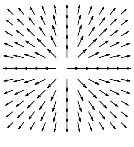
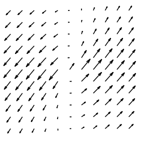
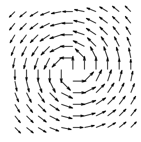
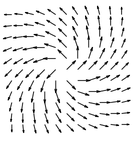
node	saddle	star node
		
$\lambda_1\lambda_2 > 0$ $\Delta > 0, \det(C) > 0$ $a^\mathsf{T} = [2,0,0,0,4,0]$	$\lambda_1\lambda_2 < 0$ $\Delta > 0, \det(C) < 0$ $a^\mathsf{T} = [2,0,0,0,1,0]$	$\lambda_1 = \lambda_2$ $\Delta = 0, c_{1,2} = c_{2,1}$ $a^\mathsf{T} = [0,0,0,0,1,0]$
improper node	center	spiral
		
$\lambda_1 = \lambda_2$ $\Delta = 0, c_{1,2} =\neq c_{2,1}$ $a^\mathsf{T} = [1,1,0,0,1,\sqrt{2}]$	$\lambda_{1,2} = \alpha \pm \beta i$ $\Delta < 0, \mathrm{tr}(C) = 0$ $a^\mathsf{T} = [0,0,0,0,0,1]$	$\lambda_{1,2} = \alpha \pm \beta i$ $\Delta < 0, \mathrm{tr}(C) \neq 0$ $a^\mathsf{T} = [0,0,0,0,1,1]$

Figure B.16: Phase portrait classification based on the eigenvalues of C.

Phase-portrait methods [18, 1] locally approximate a flow pattern $(\dot{x}, \dot{y})^\mathsf{T}$, $\dot{x}(t) = dx/dt$ using 2-D linear differential equations, and the following affine model:

$$\begin{aligned} \dot{x} &= c_{1,1}x + c_{1,2}y + c_{1,3} \\ \dot{y} &= c_{2,1}x + c_{2,2}y + c_{2,3}, \end{aligned} \tag{B.3}$$

which can be written in matrix form as $\mathbf{v} = \mathbf{Cx} + \mathbf{b}$. After locally fitting this linear model to the flow-field data, the detected singular pattern is classified according to the eigenvalues of the coefficient matrix $\mathbf{C}$. Let $\mathrm{tr}(\mathbf{C})$ be the trace of $\mathbf{C}$ and $\det(\mathbf{C})$ its determinant, the eigenvalues of $\mathbf{C}$ are given by [18, 1]:

$$\lambda_1 = \frac{1}{2}\left(\mathrm{tr}(\mathbf{C}) + \sqrt{\Delta}\right), \quad \text{and} \quad \lambda_2 = \frac{1}{2}\left(\mathrm{tr}(\mathbf{C}) - \sqrt{\Delta}\right), \tag{B.4}$$

where $\Delta = \mathrm{tr}(\mathbf{C})^2 - 4\det(\mathbf{C})$. By substituting $x = \frac{z+\bar{z}}{2}$ and $y = \frac{\bar{z}-z}{2}i$ into the affine model in (B.3), we have:

$$\begin{aligned} 2\dot{x} &= c_{1,1}(z+\bar{z}) + c_{1,2}(\bar{z}-z)i + c_{1,3} \\ 2\dot{y} &= c_{2,1}(z+\bar{z}) + c_{2,2}(\bar{z}-z)i + c_{2,3} \end{aligned} \tag{B.5}$$

To simplify the notation, we drop the constant background flows $c_{1,3}$ and $c_{2,3}$ in (B.5) because they do not affect eigenvalue calculation. We can write the flow field as a complex number $\dot{x}+\dot{y}\,i$, and compare it with the Laurent expansion to obtain:

$$\begin{aligned} \dot{x} + \dot{y}i = \frac{1}{2}[&\underbrace{(c_{1,1} + c_{2,2})}_{a_{1,1}} z + \underbrace{(c_{2,1} - c_{1,2})}_{a_{1,2}} zi + \\ &\underbrace{(c_{1,1} - c_{2,2})}_{a_{-1,1}} \bar{z} + \underbrace{(c_{1,2} + c_{2,1})}_{a_{-1,2}} \bar{z}i]. \end{aligned} \tag{B.6}$$

We used the approximation $\bar{z} \approx z^{-1}$. As a result, we have:

$$\begin{bmatrix} c_{1,1}, & c_{1,2} \\ c_{2,1}, & c_{2,2} \end{bmatrix} \approx \frac{1}{2} \begin{bmatrix} a_{1,1} + a_{-1,1}, & a_{-1,2} - a_{1,2} \\ a_{-1,2} + a_{1,2}, & a_{1,1} - a_{-1,1} \end{bmatrix} \tag{B.7}$$

with

$$\mathrm{tr}(A) = 2a_{1,1}, \qquad \det(A) = a_{1,1}^2 - a_{-1,1}^2 + a_{1,2}^2 - a_{-1,2}^2,$$

and

$$\begin{aligned} \Delta &= 4a_{1,1}^2 - 4\left(a_{1,1}^2 - a_{-1,1}^2 + a_{1,2}^2 - a_{-1,2}^2\right) \\ &= 4\left(a_{-1,1}^2 - a_{1,2}^2 + a_{-1,2}^2\right). \end{aligned}$$

This shows that the phase-portrait can be approximated using the first-order coefficients $a_{k,j}$ for $\|k\| \leq 1$ from our representation. Figure B.16 shows examples of singular patterns of different phase portraits produced using our model. Similarly, we can also establish the connection between the higher-order coefficients in our model and the non-linear phase portraits [17].

References

[1] A. R. Rao, R. C. Jain, Computerized flow field analysis: Oriented texture fields, IEEE Transaction of Pattern Analysis and Machine Intelligence 14 (7) (1992) 693–709.

[2] H. Nogawa, Y. Nakajima, Y. Sato, S. Tamura, Acquisition of symbolic description from flow fields: a new approach based on a fluid model, IEEE Transaction of Pattern Analysis and Machine Intelligence 19 (1) (1997) 58–63.

[3] L. Fan, S. Wang, H. Wang, T. Guo, Singular points detection based on zero-pole model in fingerprint images, IEEE Transaction of Pattern Analysis and Machine Intelligence 30 (6) (2008) 929–940.

[4] K. Nilsson, J. Bigun, Localization of corresponding points in fingerprints by complex filtering, Pattern Recognition Letters 24 (13) (2003) 2135–2144.

[5] T. Corpetti, E. Mémin, P. Pérez, Extraction of singular points from dense motion fields: An analytic approach, Journal of Mathematical Imaging and Vision 19 (3) (2003) 175–198.

[6] K. Y. Wong, C. L. Yip, Identifying centers of circulating and spiraling vector field patterns and its applications, Pattern Recognition 42 (7) (2009) 1371–1387.

[7] L. Xu, H. Dinh, E. Zhang, Z. Lin, R. Laramee, A distribution-based approach to tracking points in velocity vector fields, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 476–491.

[8] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.

[9] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, IEEE Transaction of Pattern Analysis and Machine Intelligence 27 (10) (2005) 1615–1630.

[10] F. H. Post, B. Vrolijk, H. Hauser, R. S. Laramee, H. Doleisch, The state of the art in flow visualization: Feature extraction and tracking, Computer Graphics Forum 22 (4) (2003) 775–792.

[11] W. Liu, E. Ribeiro, Scale and rotation invariant detection of singular patterns in vector flow fields, in: IAPR International Workshop on Structural Syntactic Pattern Recognition (S-SSPR), 2010, pp. 522–531.

[12] J. Ebling, G. Scheuermann, Clifford convolution and pattern matching on vector fields, in: IEEE Visualization, 2003, pp. 193–200.

[13] M. Schlemmer, M. Heringer, F. Morr, I. Hotz, M. Hering-Bertram, C. Garth, W. Kollmann, B. Hamann, H. Hagen, Moment invariants for the analysis of 2D flow fields, IEEE Transactions on Visualization and Computer Graphics 13 (6) (2007) 1743–1750.

[14] R. M. Ford, R. N. Strickland, Representing and visualizing fluid flow images and velocimetry data by nonlinear dynamical systems, Graphical Models and Image Processing 57 (6) (1995) 462–482.

[15] M. Kass, A. Witkin, Analyzing oriented patterns, Computer Vision, Graphics, and Image Processing 37 (3) (1987) 362–385.

[16] R. M. Ford, R. N. Strickland, B. A. Thomas, Image models for 2-D flow visualization and compression, Graphical Models and Image Processing 56 (1) (1994) 75–93.

[17] J. Li, W. Yau, H. Wang, Constrained nonlinear models of fingerprint orientations with prediction, Pattern Recognition 39 (1) (2006) 102–114.

[18] O. Kihl, B. Tremblais, B. Augereau, Multivariate orthogonal polynomials to extract singular points, in: IEEE International Conference on Image Processing, 2008, pp. 857–860.

[19] B. Sherlock, D. Monro, A model for interpreting fingerprint topology, Pattern Recognition 26 (7) (1993) 1047–1055.

[20] J. Gu, J. Zhou, D. Zhang, A combination model for orientation field of fingerprints, Pattern Recognition 37 (3) (2004) 543–553.

[21] K. Y. Wong, C. L. Yip, Identifying centers of circulating and spiraling vector field patterns and its applications., Pattern Recognition (2009) 1371–1387.

[22] J. Zhou, J. Gu, Modeling orientation fields of fingerprints with rational complex functions, Pattern Recognition 37 (2) (2004) 389–391.

[23] M. Jiang, R. Machiraju, D. Thompson, Detection and visualization of vortices, in: The visualization handbook, Academic Press, 2005, pp. 295–315.

[24] R. C. Strawn, J. Ahmad, D. N. Kenwright, Computer visualization of vortex wake systems, AIAA Journal 37(4) (1999) 511–512.

[25] Y. Levy, D. Degani, A. Seginer, Graphical visualization of vortical flows by means of helicity, AIAA journal 28 (8) (1990) 1347–1352.

[26] I. Sadarjoen, F. Post, B. Ma, D. Banks, H.-G. Pagendarm, Selective visualization of vortices in hydrodynamic flows, in: IEEE Visualization, 1998, pp. 419–422.

[27] Y. Wang, J. Hu, Rotation estimation of singular points using analytical models, in: Proceedings of SPIE, Vol. 6944, SPIE, 2008, p. 69440L.

[28] L. Florack, Scale-space theories for scalar and vector images, in: Scale-Space '01, Springer-Verlag, London, UK, 2001, pp. 193–204.

[29] C. Jin, H. Kim, Pixel-level singular point detection from multi-scale Gaussian filtered orientation field, Pattern Recognition 43 (11) (2010) 3879 – 3890.

[30] D. Weng, Y. Yin, D. Yang, Singular points detection based on multi-resolution in fingerprint images, Neurocomputing 74 (17) (2011) 3376 – 3388.

[31] M. K. Hu, Visual pattern recognition by moment invariants, IEEE Transactions on Information Theory 8 (1962) 179–187.

[32] H. Marquering, P. Van Ooij, G. Streekstra, J. Schneiders, C. Majoie, E. vanBavel, A. Nederveen, Multi-scale flow patterns within an intracranial aneurysm phantom, IEEE Transactions on Biomedical Engineering 58 (12) (2011) 3447–3450.

[33] M. Reisert, H. Burkhardt, Equivariant holomorphic filters for contour denoising and rapid object detection, IEEE Transactions on Image Processing 17 (2) (2008) 190 –203.

[34] Q. Wang, O. Ronneberger, H. Burkhardt, Rotational invariance based on fourier analysis in polar and spherical coordinates, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (9) (2009) 1715 –1722.

[35] I. Daubechies, W. Sweldens, Factoring wavelet transforms into lifting steps, Journal of Fourier Analysis and Applications 4 (1998) 247–269.

[36] J. S. Geronimo, M.-J. Lai, Factorization of multivariate positive Laurent polynomials, Journal of Approximation Theory 139 (1-2) (2006) 327 – 345.

[37] I. Bronstein, K. Semendyayev, G. Musiol, H. Muehlig, Handbook of mathematics, Springer Berlin, 1997.

[38] R. Horn, C. Johnson, Matrix analysis, Cambridge University Press, 1990.

[39] M. J. Forray, Approximation Theory and Methods, Cambridge University Press, 1981.

[40] Z. Fan, F. Qiu, A. Kaufman, S. Yoakum-Stover, GPU cluster for high performance computing, IEEE Supercomputing Conference (2004) 47.

[41] S. G. Krantz, Handbook of Complex Variables, Birkhäuser, 1999.

[42] A. Bultheel, Orthogonal rational functions, Cambridge University Press, 1999.

[43] J. Carlier, Second set of fluid mechanics image sequences, European Project 'Fluid image analysis and description' (FLUID) `http://www.fluid.irisa.fr/` (2005).

[44] Y. Li, E. Perlman, M. Wan, Y. Yang, C. Meneveau, R. Burns, S. Chen, A. Szalay, G. Eyink, A public turbulence database cluster and applications to study Lagrangian evolution of velocity increments in turbulence, Journal of Turbulence 9 (31) (2008) 1–29.

[45] W. Liu, E. Ribeiro, A novel consistency regularizer for meshless nonrigid image registration, in: International Cconference on Advances in Visual Computing, 2010, pp. 242–251.