Moment Invariants for 2D Flow Fields via Normalization in Detail

Roxana Bujack, Ingrid Hotz, Gerik Scheuermann, and Eckhard Hitzer

Abstract—The analysis of 2D flow data is often guided by the search for characteristic structures with semantic meaning. One way to approach this question is to identify structures of interest by a human observer, with the goal of finding similar structures in the same or other datasets. The major challenges related to this task are to specify the notion of similarity and define respective pattern descriptors. While the descriptors should be invariant to certain transformations, such as rotation and scaling, they should provide a similarity measure with respect to other transformations, such as deformations. In this paper, we propose to use moment invariants as pattern descriptors for flow fields. Moment invariants are one of the most popular techniques for the description of objects in the field of image recognition. They have recently also been applied to identify 2D vector patterns limited to the directional properties of flow fields. Moreover, we discuss which transformations should be considered for the application to flow analysis. In contrast to previous work, we follow the intuitive approach of *moment normalization*, which results in a complete and independent set of translation, rotation, and scaling invariant flow field descriptors. They also allow to distinguish flow features with different velocity profiles. We apply the moment invariants in a pattern recognition algorithm to a real world dataset and show that the theoretical results can be extended to discrete functions in a robust way.

Index Terms—Moments, Moment Invariants, Pattern Recognition, Flow Visualization, Normalization

✦

1 INTRODUCTION

This paper is an extended version of "Moment Invariants for 2D Flow Fields via Normalization", [3]. It describes the same method as its predecessor but in more detail and depth. The major innovations of this representation at full length are:

- Explicit description of the algorithm.
- Analysis of its runtime and memory requirements.
- Derivation of the normalization with respect to reflections.
- Extensive discussion of the considered transformations and the effects of normalization with respect to them.
- More examples and beautiful images.

Visualization and data analysis play an essential role in the process of understanding flow simulations. The definition and extraction of characteristic flow structures from the data is of special importance and is the topic of many discussions in the field of fluid mechanics. Respective questions concern, e. g., the formation and development of "coherent structures" [15], sometimes identified with vortices. Even though many scientists have an intuitive feeling about such structures, there is no commonly accepted definition. It is often challenging to

- *Roxana Bujack is with the Department of Computer Science, Leipzig University, Leipzig, Germany, ESF Application No. 100098251.*
  E-mail: bujack@informatik.uni-leipzig.de
- *Ingrid Hotz is with the Department of Simulation and Software Technology, German Aerospace Center, Braunschweig, Germany.*
  E-mail: hotz@gzib.de
- *Gerik Scheuermann is with the Department of Computer Science, Leipzig University, Leipzig, Germany.*
  E-mail: scheuermann@informatik.uni-leipzig.de
- *Eckhard Hitzer is with the Department of Material Science, International Christian University, Tokyo, Japan.*
  E-mail: hitzer@icu.ac.jp

translate these intuitive notions into a mathematically tractable property. The goal of this work is to support this task allowing flexible pattern definition, e. g. through visual selection. Thereby, the major challenge is the definition of expressive descriptors. They should be detailed enough to encode the relevant information about a pattern but also general enough to allow variations in terms of size and orientation. Once structures of interest are identified, similar patterns can be automatically detected.

Similar questions can be found in the field of image analysis. There, very successful and commonly used shape descriptors for automatic object recognition are moment invariants. Moments are characteristic numbers of a function. For example, the mean and the variance are moments. They are the projection of a function to an L^2 function space basis. They are robust, flexible, easy to use, and an excellent tool to construct invariants. Invariants mean in this context that they do not change under certain transformations. Their invariance property allows to compare objects in one single step instead of considering every possible transformed version of it. Since moments have been introduced about 50 years ago, many different categories of invariants have been developed and analyzed [11].

To achieve invariance of the descriptors, two major approaches have been proposed in the past. One way is explicitly giving a calculation rule for a set of invariants, Flusser [10]. Another way is the method of normalization [6]. Flusser et al. state that both methods are equivalent [11] for scalar fields. Some of these ideas have been generalized to two-dimensional vector fields by Schlemmer et al. [25] proposing a set of complex invariant vector moments following the first approach. The second option has not yet been generalized to the vector field case.

Besides the technical generalization of the descriptors to vector-valued fields, there are other significant conceptual changes from image pattern recognition to flow analysis. Character recognition mostly heads for a classification tolerating rigid body transformations, scaling, as well as other deformations. In contrast to that, the goal in flow analysis is to establish a notion of similarity that also supports measuring changes in the flow. For many applications, it is also important to be able to abstract from background flows. These requirements make a specific discussion of considered invariances necessary.

In this paper we deal with these conceptual questions as well as the technical generalization of moment invariants respecting flow specific transformations. We introduce a new approach to vector moments, generalizing the theory of 2D invariants from scalar functions to 2D vector fields making use of the isomorphism between the Euclidean and the complex plane.

In a nutshell the method presented in this paper works as follows:

- **Moments** are the projections of a function to a function space basis. In order to compare the functions, it is sufficient to compare their moments.
- **Normalization** is the act of transforming a function or rather their moments into a predefined standard position.
- **Moment invariants** are characteristic numbers constructed from moments that do not change under certain transformations. They can be produced by normalization. Because if all the functions are in one standard position, their prior position has no influence on their normalized moments.

2 RELATED WORK

The analysis of vector fields has a long tradition in the area of visualization. Accordingly, there has been much interesting work, which goes beyond the scope of this section. But we would like to point at some good overview articles dealing with vector field visualization with different foci: Texture and Feature-Based Flow Visualization [9], Integration-Based Geometric Flow Visualization [19], and Illustrative Flow Visualization [2].

Of special interest in context with the represented method, are *feature extraction* and *pattern recognition* methods. Typical vector features may either be directly based on the given vector field, e. g. *vector field topology*, or on derived scalar, vector, or tensor fields. Vector field topology focuses on finding features like sources, sinks, and saddle points as well as separatrices connecting them [17], [23]. Scalar features are mostly defined as iso-contours or as the extremal structures of a derived scalar field [27]. Examples are vortex like features using identifiers such as vorticity [21], [22], λ_2 [14], or the acceleration magnitude [16], all based on the Jacobian matrix of the flow field. Such predefined features are very successful when looking for specific well-known structures. But they are probably too specific when looking for more general patterns.

A more flexible way to define features interactively as *patterns* is provided by methods originating from image processing. In contrast to the features described above, such patterns are not locally defined by having a spatial extension.

A first attempt in this direction has been made by Heiberg et al. [12] who introduced a convolution operator for vector field data. This idea has been further elaborated by Ebling et al. [8], [7]. To find patterns of different size and orientation, the respective filter masks have to be adjusted and the filtering process has to be performed multiple times.

To avoid these high computational costs, pattern descriptors that are invariant under rotation and scaling have been proposed. In the area of image processing, Hu [13] introduced his famous seven moment invariants to the pattern recognition community. These are expressions that do not change under shift, rotation, and scale and therefore help to identify the same object aligned differently. They are one of the most important sets of shape descriptors. There has been much related work since. The use of complex moments [28], [1] simplified the construction of rotation invariants because of the easy way to describe rotations by means of complex exponentials. Two major ways for the construction of invariants have been introduced. Flusser [10] uses an independent basis by explicitly defining a set of invariants. A different approach to achieve invariance is the method of *normalization* [6], there the pattern is brought into a standard position by setting certain moments to given values. Flusser et al. state that both methods are equivalent. For a more comprehensive introduction to moment invariants we recommend [11]. Building on this work, Schlemmer et al. [25], [24] have defined a moment basis for vector fields. Accordingly, the scale invariance is implemented by a moment pyramid, which serves as basis for an efficient comparison. These moments have then been applied to follow characteristic patterns in time-dependent datasets [26]. While generating first promising results, a concise mathematical formulation of vector moments is still missing. Another interactive feature or pattern selection method for vector fields that also considers neighborhood characteristics has been presented by Daniels et al. [5]. They define features by attributes that describe the neighborhood of a sample within the input vector field.

3 BASICS - MOMENTS FOR SCALAR FIELDS

In the following section, we summarize the most important basics for classical complex moment invariants, on which our work builds. In particular to motivate our design decision, we discuss the two different approaches to produce invariant descriptors: the construction of an invariant basis in comparison to normalization.

Throughout the paper, we will perform all theoretical calculations in the notation of complex numbers. Please keep in mind that every result for a complex function $f : \mathbb{C} \to \mathbb{C}$

$$f(z) = f_1(x_1 + ix_2) + if_2(x_1 + ix_2) \simeq \begin{pmatrix} v_1(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}) \\ v_2(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}) \end{pmatrix} = v(x) \tag{1}$$

can be automatically understood as a result for a two-dimensional vector field $v : \mathbb{R}^2 \to \mathbb{R}^2$ using the isomorphism with $v_{1/2} = f_{1/2}$.
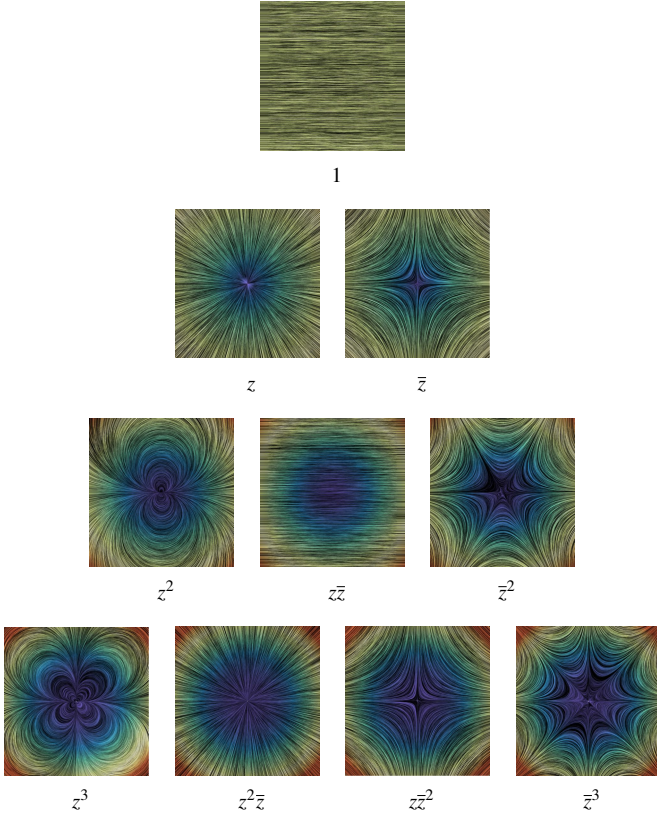
Fig. 1. The first complex monomials with coefficient $1 \in \mathbb{R}$ interpreted as 2D vector fields visualized with line integral convolution (LIC) [4] and a color map representing the velocity. Blue means low and red high velocity, compare colorbar in Figure 6.
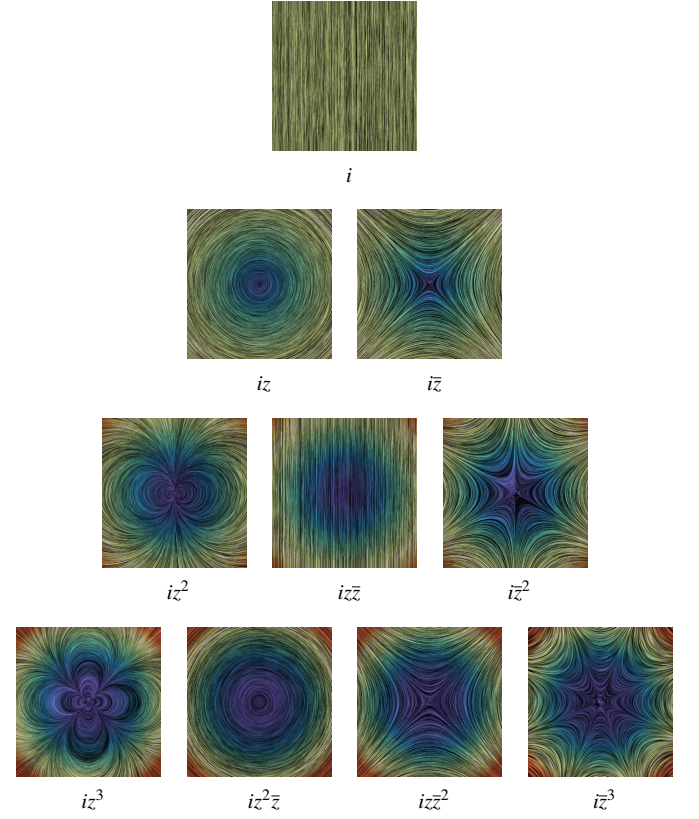

Fig. 2. The first complex monomials with coefficient $i \in \mathbb{C}$ interpreted as 2D vector fields visualized with line integral convolution (LIC) [4] and a color map representing the velocity. Blue means low and red high velocity, compare colorbar in Figure 6.

3.1 Complex moments

The moments of a scalar field or function are its projections to a function space basis. We are dealing with functions defined over $\mathbb{R}^2 \simeq \mathbb{C}$ and use complex moments [28], [1], [18], which correspond to the standard complex monomials $z^p \bar{z}^q$. The first complex monomials interpreted as 2D vector fields are shown in Figure 1. Complex moments are easy to use, interpret, and implement and sufficiently powerful for our purpose. Further, the polynomial space is dense in the space of the continuous functions, which makes its reduction onto a compact area Ω dense in the space of the square integrable functions $L_2(\Omega)$. The moments are only the coefficients with respect to a basis, if the basis is orthonormal, which the monomials are not. But the coefficients and the projections behave equally under the considered transformations. The complex moments were originally introduced to deal with real valued functions, but the generalization to complex-valued functions is straight forward.

Definition 1. *For the pair $p, q \in \mathbb{Z}$, with grade $n = p + q$ and the complex function $f : \mathbb{C} \to \mathbb{C}$ with compact support, the* **complex moments** $c_{p,q}$ *are defined as*

$$c_{p,q} = \int_{\mathbb{C}} z^p \bar{z}^q f(z) \, dz. \tag{2}$$

On a compact set $\Omega \subset \mathbb{C}$, the moments are the projections of the vector field onto the basis functions, which are calculated using the $L_2(\Omega)$ scalar product $c_{p,q} = \langle z^q \bar{z}^p, f(x) \rangle_2$. In practical applications, all fields have a compact support.

Using the polar form for complex numbers $z = r e^{i\phi} \in \mathbb{C}$, we can alternatively write

$$c_{p,q} = \int_0^1 \int_0^\infty r^{p+q} e^{i\phi(p-q)} f(r, \phi) r \, d\phi \, dr. \tag{3}$$

The complex moments of low orders have a very intuitive geometric meaning. The zeroth order moment

$$c_{0,0} = \int_{\mathbb{C}} f(z) \, dz \tag{4}$$

can be interpreted as the mass of the function. The moments of order one represent the center of mass of a real valued function via

$$\frac{c_{1,0}}{c_{0,0}} = \frac{\int_{\mathbb{C}} z f(z) \, dz}{\int_{\mathbb{C}} f(z) \, dz}. \tag{5}$$

Example 1. *To illustrate the geometric meaning of the moments, we use the characteristic function $f : \mathbb{C} \to \{0, 1\}$ representing the triangle in Figure 3 (a) as an example,*

$$f(z) = \begin{cases} 1, & \text{if } 0 < \text{Re}(z) < 1 \text{ and } 0 < \text{Im}(z) < \text{Re}(z), \\ 0, & \text{else.} \end{cases} \tag{6}$$

Its moments up to the second order are

$$c_{0,0} = \frac{1}{2}, \quad c_{1,0} = \frac{1}{3} + \frac{1}{6}i, \quad c_{0,1} = \frac{1}{3} - \frac{1}{6}i,$$
$$c_{1,1} = \frac{1}{3}, \quad c_{2,0} = \frac{1}{6} + \frac{1}{4}i, \quad c_{0,2} = \frac{1}{6} - \frac{1}{4}i. \tag{7}$$

The surface area or mass of the triangle is given by the zeroth order moment $c_{0,0} = 1/2$, and its center of mass by the first order moment $c_{1,0}/c_{0,0} = 2/3 + 1/3i$.

3.2 Moment invariants

Useful descriptors on the basis of moments should have invariance properties. In general invariants are characteristics that do not change under certain transforms. Depending on the specific application, interesting transforms can be changes in position, size, orientation, convolution, affine transforms, blur, perspective, contrast, or color.

To fulfill the demand for invariances, two basically different approaches have been proposed. These are:

- Construction of a basis of moment invariants.
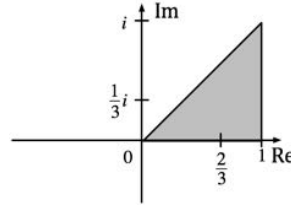- Normalization of the moments.

According to Flusser et al. [11], these approaches may have different origins but are equivalent with respect to their results. The first approach defines an explicit calculation rule for an independent and complete basis. Applying this rule, an infinite set of moment invariants can be generated. The calculation rules are usually inspired by results of the much older field of algebraic invariants and are not very intuitive. The second approach, which is called *normalization*, is much easier to imagine. In order to achieve an invariant description of the patterns, a standard position is defined. The easiest way is to set certain moments to predefined values. These chosen moments will take the same values for any pattern; all the remaining moments can be used as independent discriminators. Whenever two patterns shall be compared, there is no need to test all orientations, but only the moments of the patterns in standard position.

Example 2. *To illustrate the geometric interpretation of normalization we use again the function of Example 1 and define a standard position with respect to translation, scaling, and rotation.*
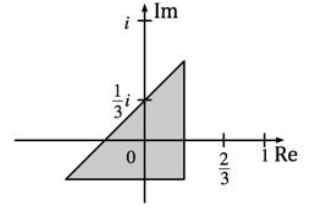
- *Translation: a self-evident standard with respect to translation would be the claim for the center of mass to coincide with the origin of coordinates. In the language of moments, that means we set the moment $c_{1,0} = 0$.*
- *Scaling: a reasonable suggestion is to demand the area of the pattern to have unit magnitude, i. e. $c_{0,0} = 1$.*
- *Rotation: in order to standardize the orientation of a pattern, we can choose a moment and align it with the positive real axis. Usually the moment $c_{2,0} \in \mathbb{R}^+$ is chosen.*

The shape of the triangle after every step can be followed in Figure 3. The normalized moments of the triangle are
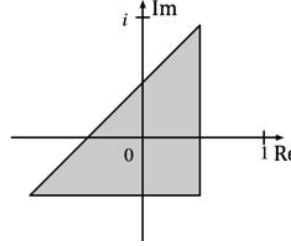
$$c_{0,0} = 1, \quad c_{1,0} = 0, \quad c_{0,1} = 0,$$
$$c_{1,1} = \frac{2}{9}, \quad c_{2,0} = \frac{1}{9}, \quad c_{0,2} = \frac{1}{9}. \tag{8}$$
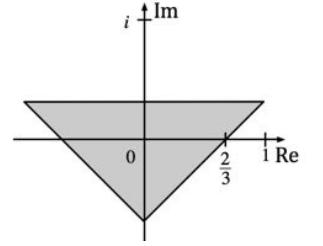

The original triangle from (6) — Normalization with respect to translation

Normalization with respect to translation and scaling — Normalization with respect to translation, scaling, and rotation

Fig. 3. Normalization of the triangle from equation (6)

Please note that other choices for a standard position would lead to equally valid normalizations. This one coincides with aligning the principal axes of the principal component analysis to the Cartesian basis axes.

In practice, the normalization process is not done by explicitly moving the pattern. To describe and compare different patterns, it is sufficient to normalize the moments. Thus, no resampling and interpolation of the function is necessary. Normalization has many advantages compared to the independent basis approach.

- It has a clear motivation and reasonable geometric interpretation.
- No work needs to be put into the analysis and proof of the independence and the completeness because these properties are directly inherited from the function space basis.
- Its generalization to higher dimensions and other kinds of functions and spaces is straightforward.

It should be noted that normalization cannot be used to create invariants with respect to a transform that has no reasonable standard representation, like blur. Since our objective is invariance with respect to translation, rotation, and scaling, this is no issue for our application. Another disadvantage is a certain instability in the special situation when a function is at the very threshold such that a rotation to the left and a rotation to the right are equally suitable to achieve the standard position. Due to the prevalence of the advantages of the normalization approach for flow pattern recognition, we decided to follow this approach.

4 MOMENT INVARIANTS FOR FLOW FIELDS

In this section, we discuss moment invariants applied to pattern analysis for flow fields. Many of the ideas introduced for shape recognition can be generalized but there are also

substantial differences.

Relevant transformations – An essential decision is the class of transformations that are considered for invariance. There are many more options to define geometric transformations for vector fields than for scalar functions and different transformations are of significance. To compare patterns with arbitrary orientation, position, and size, it is not sufficient to apply the transformation to the domain. It is necessary to transform the vectors correspondingly. In the following, we refer to the transformation of the domain as *inner transformation* and the change of the values of the vector field as *outer transformation*.

Driving questions – Moreover, the driving questions are very different. In shape analysis, the questions are often related to a discrete classification of pre-segmented patterns, whereas in flow analysis, we are interested in a similarity measure that expresses the strength of a given feature at a certain position. Relevant patterns are often relatively small compared to the size of the field and can even exist at the same position at different scales.

For a general complex function, translation, rotation, and scaling can be applied to its argument and its value. That means we generally deal with eight degrees of freedom

$$f'(z) = s_o e^{i\alpha_o} \big(f(s_i e^{i\alpha_i} z + t_i) + t_o \big), \tag{9}$$

with the inner and outer scaling factors $s_i, s_o \in \mathbb{R}^+$, translational differences $t_i, t_o \in \mathbb{C}$, and rotation angles $\alpha_i, \alpha_o \in [-\pi, \pi]$. In the following we will discuss these six central transformations.

Rotations

Since the rotation invariance is of special importance in the context of flow analysis, we will describe this transformation in more detail. Examples for rotations of a vector field are shown in Figure 4. Analogous considerations are also valid for other geometric transformations such as translation and scaling.
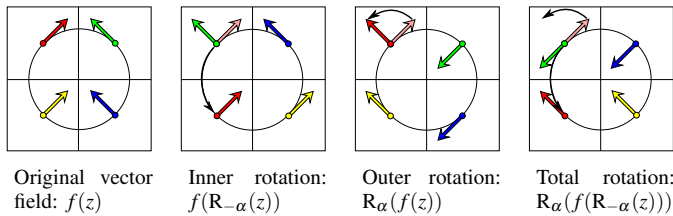

Original vector field: $f(z)$ | Inner rotation: $f(\mathrm{R}_{-\alpha}(z))$ | Outer rotation: $\mathrm{R}_\alpha(f(z))$ | Total rotation: $\mathrm{R}_\alpha(f(\mathrm{R}_{-\alpha}(z)))$

Fig. 4. Effect of the rotation operator R_α applied to an examplary vector field in three different ways.

Let R_α be an operator that describes a mathematically positive rotation by the angle α and let $f, f' : \mathbb{C} \to \mathbb{C}$ be two vector fields. We say the two fields differ by an *inner rotation* if

$$f'(z) = f(\mathrm{R}_{-\alpha}(z)). \tag{10}$$

This means that the starting position of every vector is rotated by α and then the original vector is reattached at the new position. The inner rotation is suitable to describe the rotation of a 2D color image or a complex valued function over a plane. The color or the complex value respectively is represented as a vector and does not change when the underlying domain is rotated. The two vector fields differ by an *outer rotation* if

$$f'(z) = \mathrm{R}_\alpha(f(z)). \tag{11}$$

Here, every vector on the vector field f' is a rotated copy of every vector in the vector field f, while its location remains fixed. For complex valued functions, it describes a phase shift in the image space. This kind of rotation appears, for example, in color images when the color space is turned but the picture is not moved [20]. A third type is the *total rotation*, which combines the inner and outer rotation

$$f'(z) = \mathrm{R}_\alpha(f(\mathrm{R}_{-\alpha}(z))). \tag{12}$$

It represents a coordinate transform for a vector field with geometric, physical meaning, like a flow field. Here the positions and the vectors are stiffly connected during the rotation. This is the kind of rotation, we will use in this paper.

Reflections

Reflections and rotations are very closely related. Considering flow fields, also the total reflections are the ones of interest. They can be produced from a total rotation followed by reflection along the real axis, which is a complex conjugation

$$f'(z) = \overline{\mathrm{R}_\alpha(f(\mathrm{R}_{-\alpha}(\bar{z})))}. \tag{13}$$

This property makes the treatment of reflections almost trivial. Therefore, we do not explicitly include them into the following calculations.

Scaling and Translation

Because flow patterns have a limited spatial extend, we do not want to compare fields but only parts of them. This means, we have to restrict the analysis to windows of the size of the pattern. Thus, the inner translation and scaling cannot be covered using moment invariants. This problem is solved by searching at 'all' possible places and for 'all' possible scales in the big vector field. As a result, it is not useful to include these parameters in the calculation (9) we set $t_i = 0, s_i = 1$. To be in accordance with rotation invariance, we have chosen a circular window $A = B_r(0)$.

The outer translation can be interpreted as a distortion of the pattern by some background flow or a moving frame of reference. Since we would like to be able to detect moving flow patterns, we will consider normalization with respect to outer translation t_o. The outer scale represents the velocity of the flow. We want to detect the pattern independent from its speed, so we also normalize with respect to outer scale s_o. Please note that during this operation we will not set every vector to unit length. The ratio between the lengths of the vectors and the velocity pattern are preserved.

In Summary: Considered Transformations

All in all, the transforms of a function $f(z)$ with respect to which we want to normalize, take the shape

$$f'(z) = se^{i\alpha}\left(f(e^{-i\alpha}z) + t\right), \tag{14}$$

with the scaling factor $s \in \mathbb{R}^+$, translational difference $t \in \mathbb{C}$, rotation angle $\alpha \in [-\pi, \pi]$. In the next section, we will show how this special kind of normalization can be produced.

Discrete Formulation

For the practical computation a discrete formulation of the integral definitions in Sec. 3.1 has to be used. For a given position $z_0 = x_0 + iy_0$ and scale s, discrete functions are sampled on a uniform grid with spacing $h = 1/s$ and the moments of order $n = p + q$ are computed as

$$c_{pq}(z_0) = \sum_{k,l=-s}^{s} \sum_{l=-\sqrt{s^2+k^2}}^{\sqrt{s^2+k^2}} (kh + ilh)^p (kh - ilh)^q f(z_0 + kh + ilh). \tag{15}$$

It should be noted that integration using discrete filters, reduces the accuracy of the rotation invariance, see Section 8. The computation of the moments is defined as a convolution and can be efficiently performed using the fast Fourier transform (FFT).

5 CONSTRUCTION OF THE INVARIANTS BY NORMALIZATION

The transformation (14) has four real, respectively two complex, degrees of freedom. This means, in order to define a standard position with respect to total rotation, outer scaling, and outer translation for the normalization, we have to choose two complex moments and move the function such that these are set to specified values. These moments should be of low order to be robust [1]. Mathematically speaking, we look for parameters $s_0 \in \mathbb{R}^+, t_0 \in \mathbb{C}$, and $\alpha_0 \in [-\pi, \pi)$, such that the function

$$f^0(z) = s_0 e^{i\alpha_0}\left(f(e^{-i\alpha_0}z) + t_0\right) \tag{16}$$

has two complex moments with fixed values.

Lemma 1. *Let $s \in \mathbb{R}^+, t \in \mathbb{C}, \alpha \in [-\pi, \pi)$ be parameters for outer scaling, outer translation, and total rotation and let*

$$f'(z) = se^{i\alpha}\left(f(e^{-i\alpha}z) + t\right), \tag{17}$$

be the transformed copy of a complex function $f : \mathbb{C} \to \mathbb{C}$. Then, the complex moments $c'_{p,q}$ of f' over the circular, compact area $A = B_r(0)$ satisfy

$$c'_{p,q} = se^{i\alpha(p-q+1)}\left(c_{p,q} + t\int_A z^p \bar{z}^q \, dz\right). \tag{18}$$

Proof: With a suitable substitution of the integration variable, the complex moments $c'_{p,q}$ of f' suffice

$$
\begin{aligned}
c'_{p,q} &= \int_A z^p \bar{z}^q f'(x,y) \, dz = \int_A z^p \bar{z}^q se^{i\alpha}\left(f(e^{-i\alpha}z) + t\right) dz \\
&= se^{i\alpha} \int_A (e^{i\alpha}z)^p (\overline{e^{i\alpha}z})^q \left(f(z) + t\right) dz \\
&= se^{i\alpha(p-q+1)} \int_A z^p \bar{z}^q \left(f(z) + t\right) dz \\
&= se^{i\alpha(p-q+1)}\left(c_{p,q} + t\int_A z^p \bar{z}^q \, dz\right),
\end{aligned}
\tag{19}
$$

which proves the assertion. $\square$

The choice of the moments that can be used for the normalization is not arbitrary. As can be seen from Lemma 1, the parameter t only has influence on moments $c'_{p,q}$ with $p = q$, because $\int_A z^p \bar{z}^q \, dz = 0$ for any pair $p \neq q$. That means we have to take one of these. A reasonable choice is setting $c_{0,0} = 0$ because in our application, the moment of order zero represents the average flow or the background flow of the field and a suitable standard position is a vanishing background flow. Applying Lemma 1 gives

$$c^0_{0,0} = se^{i\alpha_0}\left(c_{0,0} + t_0 \int_A \, dz\right) \tag{20}$$

and leads to the following condition for t_0

$$c^0_{0,0} = 0 \Leftrightarrow t_0 = -\frac{c_{0,0}}{\int_A \, dz}. \tag{21}$$

This operation is generally defined for any non vanishing area $\emptyset \neq A \subset \mathbb{C}$. So, we can always preset to zero the moment of order zero to normalize with respect to outer translation.

A classical choice for the preset value for a standard position with respect to scaling is to require unit magnitude for a selected moment. For the standard position with respect to rotation, we follow a common choice and align a moment to the positive real axis. The magnitude and the direction can both be encoded in a single complex moment. Thus, it is sufficient to choose one moment combining the normalization of rotation and scaling, and set it to one. It should be noted that a moment only qualifies as candidate for this normalization if it is nonzero. This means that the choice of an appropriate moment depends on the respective pattern function. We suggest to test the magnitude of the rotationally variant moments of the pattern in ascending order and take the first one with a significant value. We denote it by c_{p_0,q_0}. This leads to the following theorem, a main result of this paper.

Theorem 1. *Let $f : \mathbb{C} \to \mathbb{C}$ be a complex function with compact support and the complex moment $c_{p_0,q_0} \neq 0$ for a pair $p_0, q_0 \in \mathbb{N}, q_0 - p_0 \neq 1$. Then, there are $p_0 - q_0 + 1$ total rotations by angles $\alpha_0 \in [-\pi, \pi)$ and a unique outer scaling by the factor $s_0 \in \mathbb{R}^+$ such that the moment $c^0_{p_0,q_0}$ of the normalized function $f^0(z) = se^{i\alpha_0}f(e^{-i\alpha_0}z)$ takes the value 1. These are the rotations about the angles*

$$\alpha_0 = \frac{2k\pi - \arg(c_{p_0,q_0})}{p_0 - q_0 + 1} \tag{22}$$

with $k \in \mathbb{Z}$ such that $\alpha_0 \in [-\pi, \pi)$ and the scaling by the factor

$$s_0 = \frac{1}{|c_{p_0,q_0}|}. \tag{23}$$

Proof: Application of Lemma 1 gives the relation

$$c^0_{p_0,q_0} = s_0 e^{i\alpha_0(p-q+1)} c_{p_0,q_0}, \qquad (24)$$

which leads to

$$
\begin{aligned}
|c^0_{p_0,q_0}| = 1 &\Leftrightarrow |s_0 e^{i\alpha_0(p_0-q_0+1)} c_{p_0,q_0}| = 1 \\
&\Leftrightarrow |s_0||c_{p_0,q_0}| = 1 \\
&\Leftrightarrow s_0 = \frac{1}{|c_{p_0,q_0}|}
\end{aligned}
\qquad (25)
$$

and

$$
\begin{aligned}
c^0_{p_0,q_0} \in \mathbb{R}^+ &\Leftrightarrow s_0 e^{i\alpha_0(p_0-q_0+1)} c_{p_0,q_0} \in \mathbb{R}^+ \\
&\Leftrightarrow \arg(s_0 e^{i\alpha_0(p_0-q_0+1)} c_{p_0,q_0}) = 0 \\
&\Leftrightarrow \alpha_0(p_0 - q_0 + 1) + \arg(c_{p_0,q_0}) = 2k\pi \\
&\Leftrightarrow \alpha_0 = \frac{2k\pi - \arg(c_{p_0,q_0})}{p_0 - q_0 + 1}.
\end{aligned}
\qquad (26)
$$

with $k \in \mathbb{Z}$. Please note that the restriction of $s_0 \in \mathbb{R}^+$ guarantees the uniqueness of s_0 and $\alpha_0 \in [-\pi, \pi)$ the total number of $p_0 - q_0 + 1$ solutions for α_0. The existence of s_0 is ensured by the claim $c_{p_0,q_0} \neq 0$ and the existence of α_0 by the claims $q_0 - p_0 \neq 1$ and $c_{p_0,q_0} \neq 0$. $\qquad \square$

The application of these parameters to the general formula (16) gives the normalized function. The calculation of the function is not necessary. The pattern recognition is done by comparing the moments of the pattern to the ones in the field. That means we only have to transform the moments as in Lemma 1 but do not need to resample and interpolate the function.

If normalization with respect to reflections is desired, there are two possibilities. We either choose another moment with non vanishing imaginary part and demand that to be positive by applying the complex conjugation to all the moments. Or we store the moments of the pattern and their complex conjugates for comparison. The parameters for the other transforms can take infinitely many different values, while the parameter for the reflection can only take two values. Either we have to conjugate or we do not. Because of that, the increase in computational complexity for the latter method is marginal and it makes the process more flexible. We only need one moment that does not vanish and not two. Therefore we used the latter option in our final implementation.

Corollary 1. *Let $f : A = B_r(0) \to \mathbb{C}$ be a complex function with compact support and the complex moment $c_{p_0,q_0} \neq 0$ for a pair $p_0, q_0 \in \mathbb{N}, q_0 - p_0 \neq 1$. Further let*

$$t_0 = -\frac{c_{0,0}}{\int_A \mathrm{d}z}, s_0 = \frac{1}{|c_{p_0,q_0}|}, \alpha_0^k = \frac{2k\pi - \arg(c_{p_0,q_0})}{p_0 - q_0 + 1} \qquad (27)$$

with $k = 1, ..., |p_0 - q_0 + 1|$. Then, for $p, q \in \mathbb{N}$, the set of $2|p_0 - q_0 + 1|$ normalized complex moments

$$C^0_{p,q} = \{c^{0,k}_{p,q}, k = 1, ..., 2|p_0 - q_0 + 1|\} \qquad (28)$$

with

$$c^{0,k}_{p,q} = s_0 e^{i\alpha_0^k(p-q+1)} \left(c_{p,q} + t_0 \int_A z^p \bar{z}^q \, \mathrm{d}z \right), \forall k = 1, ..., |p_0 - q_0 + 1|,$$

$$c^{0,k}_{p,q} = \overline{c^{0,k-|p_0-q_0+1|}_{p,q}}, \forall k = |p_0 - q_0 + 1| + 1, ..., 2|p_0 - q_0 + 1| \qquad (29)$$

Example $f(z) = -i\bar{z} + 1 + 0.5i$ with $c_{0,0} = \pi + \frac{i\pi}{2}$, $c_{1,0} = -\frac{i\pi}{2}$

Normalized with respect to background flow $c_{0,0} = 0$: $f(z) = -i\bar{z}$

Normalized with respect to velocity $|c_{1,0}| = 1$: $f(z) = -\frac{2i\bar{z}}{\pi}$

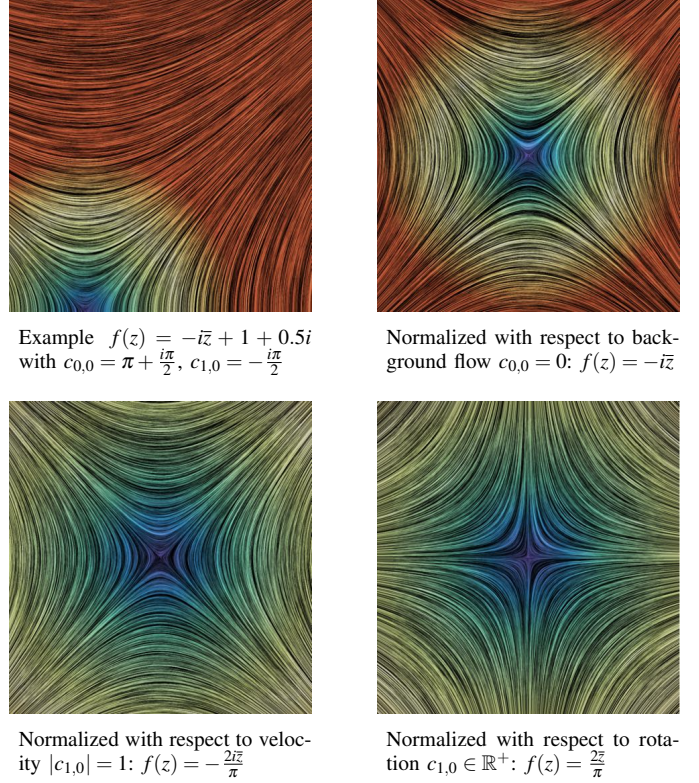Normalized with respect to rotation $c_{1,0} \in \mathbb{R}^+$: $f(z) = \frac{2\bar{z}}{\pi}$

Fig. 5. Normalization of an example saddle pattern. Blue means low and red high velocity, compare colorbar in Figure 6.

is well defined and invariant with respect to outer scaling, outer translation, total rotation, and total reflection.

Alternatively, we can reduce the set to $|p_0 - q_0 + 1|$ elements if we drop the sign of the imaginary part of all its elements.

To illustrate the normalization process applied to a flow field, we use a simple saddle pattern defined over the unit sphere as an example in Figure 5.

6 ALGORITHM AND RUNTIME

In the following section, we describe the algorithm implementing the moment normalization to detect patterns in flow fields. The input of our pattern detection algorithm is a flow field of size $N_x \times N_y$ and a search pattern g. Further parameters that have to be defined are the maximum order n of moments that are considered for the comparison and a scale discretization $\{s_1, ..., s_{N_s}\}$. The first step is to compute the moments of the pattern, see Algorithm 1 line 2. To achieve high robustness, we choose the moment with highest magnitude for the normalization in Algorithm 1 line 4. The normalization as described in Corollary 1 can be found from lines 5 to 11. The second step pre-processes the flow field. This means, we calculate the complex moments for a discrete number of positions and scales in the field to cover the inner translation and scaling invariance. Then, we normalize the moments according to Corollary 1. The third step finally computes a similarity field $S(x,s)$ for the search pattern. As similarity measure, we use the reciprocal of the minimum of the Euclidean distances of

Algorithm 1 Flow field pattern recognition

Input: $N_x \times N_y$ flow field: f, $B_r(0)$ pattern: g, maximum moment order: n, scales: $\{s_1,..,s_{N_s}\}$,

1: **for** $p+q \le n$ **do**
2:     moments of the pattern: $c^g_{p,q} = \int_{B_r(0)} z^p \bar{z}^q g(z)\, \mathrm{d}z$,
3: **end for**
4: $p_0, q_0 = \mathrm{argmax}_{p-q \ne -1, p+q \le n} |c^g_{p,q}|$,
5: $t_0 = -\dfrac{c^g_{0,0}}{\int_{B_r(0)} \mathrm{d}z}$,
6: $s_0 = \dfrac{1}{|c^g_{p_0,q_0}|}$,
7: **for** $k = 1,.., |p_0 - q_0 + 1|$ **do**
8:     $\alpha^k_0 = \dfrac{2k\pi - \arg(c^g_{p_0,q_0})}{p_0 - q_0 + 1}$,
9:     **for** $p+q \le n$ **do**
10:         normalize: $c^{g0,k}_{p,q} = s_0 e^{i\alpha^k_0 (p-q+1)} \left(c^g_{p,q} + t_0 \int_{B_r(0)} z^p \bar{z}^q\, \mathrm{d}z \right)$,
11:         normalize: $c^{g0,k+|p_0-q_0+1|}_{p,q} = \overline{c^{g0,k}_{p,q}}$,
12:     **end for**
13: **end for**
14: **for** $x \in N_x \times N_y, s = s_1,..,s_{N_s}$ **do**
15:     **for** $p+q \le n$ **do**
16:         moments of the field: $c^f_{p,q}(x,s) = \int_{B_s(x)} z^p \bar{z}^q v(z)\, \mathrm{d}z$,
17:     **end for**
18:     $t_0 = -\dfrac{c^f_{0,0}(x,s)}{\int_{B_s(0)} \mathrm{d}z}$,
19:     $s_0 = \dfrac{1}{|c^f_{p_0,q_0}(x,s)|}$,
20:     $\alpha_0 = \dfrac{-\arg(c^f_{p_0,q_0}(x,s))}{p_0 - q_0 + 1}$,
21:     **for** $p+q \le n$ **do**
22:         normalize: $c^{f0}_{p,q}(x,s) = s_0 e^{i\alpha_0 (p-q+1)} \left(c^f_{p,q}(x,s) + t_0 \int_{B_s(x)} z^p \bar{z}^q\, \mathrm{d}z \right)$,
23:     **end for**
24:     $S(x,s) = \dfrac{1}{(\min_{k=1,..,2|p_0-q_0+1|} \sum_{p+q \le n} (c^{f0}_{p,q}(x,s) - c^{g0,k}_{p,q})^2)}$,
25: **end for**

Output: similarity of the pattern p to the field f at position x and scale s: $S(x,s)$.

---

the set of moment invariants up to a given grade in Algorithm 1 line 24.

Algorithm 2 describes the visualization of the resulting three-dimensional (position and scale) scalar similarity field $S: \mathbb{R}^2 \times \mathbb{R}^+ \to \mathbb{R}$, which is the output of Algorithm 1. This visualization was done by extracting the local maxima with values above the average similarity as a threshold, compare Algorithm 2 line 3. As can be seen in line 4, for any of these local maxima, we draw a circle in the two-dimensional image plane in the following way:

- The size (scale) is represented by the diameter of the circle.
- The position (translation) is represented by its center.
- The similarity is represented by the color of the circle: red is average, yellow is high, and white is extremely high.

The runtime of the algorithm is dominated by the calculation of the moments on the flow field. It depends on the size of the field $N_x \times N_y$, the size of the pattern given by its radius r,

Algorithm 2 Visualization of the similarity

Input: similarity matrix of the $B_r(0)$ pattern g to the $N_x \times N_y$ field f at scales: $\{s_1,...,s_{N_s}\}$,

1: $S_{av} = \sum_{x \in N_x \times N_y, s=s_1,...,s_{N_s}} S(x,s) / N_x / N_y / N_s$,
2: **for** $x \in N_x \times N_y, s = s_1,..,s_{N_s}$ **do**
3:     **if** $S(x,s) > S_{av}$ **and** $\forall y \in B_1(x), \forall t \in B_1(s) : S(x,s) \ge S(y,t)$ **then**
4:         draw circle $C_s(x)$ around x with radius s in color $S(x,s)$,
5:     **end if**
6: **end for**

Output: graphic of the similarity of the pattern p to the field f.

---

which determines the integration area, the number of different scales N_s, and the maximum order of moments n. In a straight forward approach, this leads to $O(N_x N_y r^2 N_s n^2)$ operations. Our current implementation does not focus on an optimal performance. The computation of the moments corresponds to a convolution and can be efficiently implemented using a fast Fourier transform algorithm of $O(N_x N_y \log(N_x N_y) N_s n^2)$. The work of Flusser [11] considers further options for runtime improvements, which we will further explore in future work.

Compared to the time consuming moment calculation, the normalization and the comparison of the moments with $O(N_x N_y N_s n^2)$ is very fast. Therefore, we separate these two parts in practical applications. The computation of the moments can be done offline in a pre-processing step. Using the pre-computed, saved moments allows an interactive querying and exploration of the data set in real time.

For our example field, with 400×100 points, the runtime for the computation of all moments up to third order for a pattern scale of 30×30 points as in Figure 16 is 100 seconds, which we still consider feasible. The time consumption for the normalization of the moments and the visualization is negligible. It takes 0.2 seconds. Even though, our algorithm can be parallelized very well, this is the computation time for sequential computation.

In order to store the moments, the memory requirements are $O(N_x N_y N_s n^2)$.

7 SELECTION OF THE TRANSFORMATIONS

In this section, we analyze the influence of the normalization with respect to the three transformations separately. Depending on the application, it may be of advantage not to normalize with respect to all of them. Our implementation contains the option for the user to select the transformations with respect to which he wants the moments to be invariant.

As an example to illustrate these situations we used a pair of saddles in an analytic flow field of order two

$$f(z) = (\bar{z} - 1)(\bar{z} + 1) = \bar{z}^2 - 1. \tag{30}$$

In this field reduced to $[-2,2] \times [-2i, 2i] \subset \mathbb{C}$, we look for a classical first order saddle pattern

$$f(z) = i\bar{z} \tag{31}$$

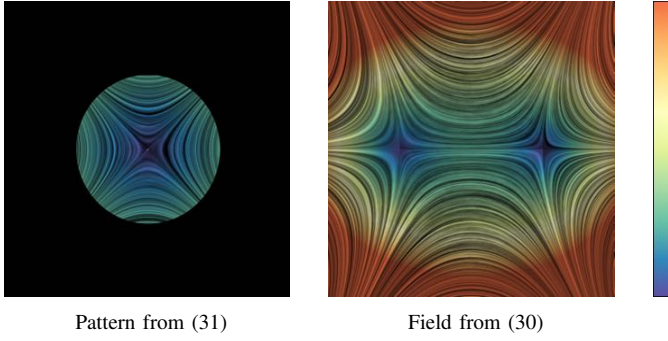Pattern from (31)   Field from (30)

Fig. 6. The query: find a saddle in the double saddle structure. The color represents the velocity of the field.

defined in the complex unit sphere. Both can be seen in Figure 6 depicted in correct ratio of their sizes. The moments are calculated in the field at discrete positions with a resolution of 0.1 and with scales $0.2, 0.3, ..., 2$.
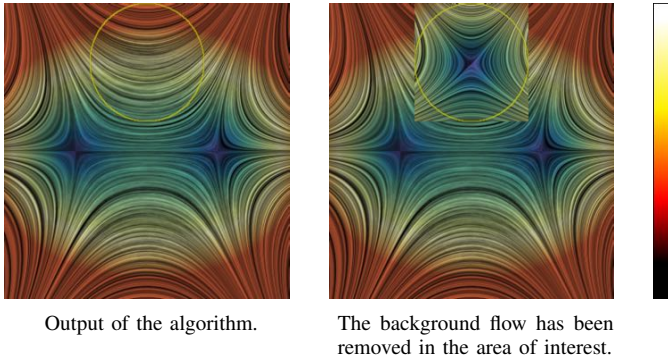

Output of the algorithm.   The background flow has been removed in the area of interest.

Fig. 7. Output of the algorithm if normalization is only performed w. r. t. translation is laid over the LIC from Figure 6. As can be seen in the color bar, low similarity is depicted by red, intermediate by yellow, and high by white circles.

First we analyze the normalization with respect to translation in Figure 7. For the purpose of illustration, we removed the mean flow of the area with the highest similarity in the image on the right. That way we can see, why the output appears at a rather surprising position. In fact, the field resembles a saddle almost everywhere, if we normalize with respect to translation. If we remove the mean flow globally, the field is the monkey saddle $f(z) = \bar{z}^2$, which can be seen in Figure 2. The area of the best fit is a little smaller than the one of the pattern. That compensates for the higher velocity in the double saddle field. The analogous spot on the lower half of the field is not a match, because there the flow goes into the opposite direction.

The result of the algorithm for normalization with respect to rotation only is pretty much as expected, compare Figure 8 top left. The two rotated and slightly distorted saddles are detected. The only interesting fact is that the similarity ($sim = 2.4$) is lower there than the one for the query with translation only ($sim = 3.4$).

The fact that the field resembles a saddle almost everywhere

Rotation   Rotation and translation

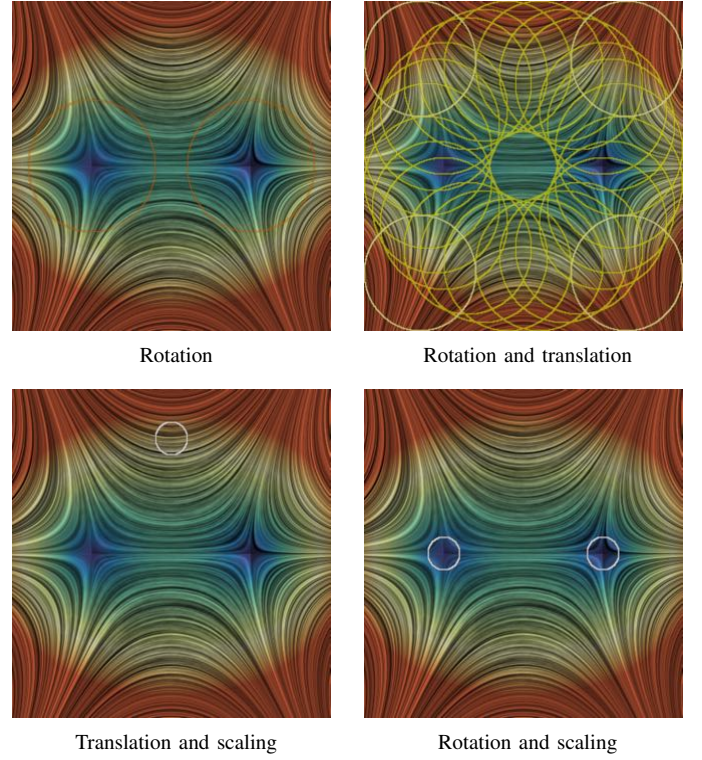

Translation and scaling   Rotation and scaling

Fig. 8. Output of the algorithm, if normalization is performed w. r. t. different combinations of transformations.

becomes very apparent, if we normalize with respect to translation and rotation. The maxima appear along the radial distance to the origin where the velocity of the pattern best matches the velocity of the higher order field, compare the top right of Figure 8. These circular maxima repeat further away from the center for smaller integration areas that even out the higher velocity of the field. Parts of the second layer can be seen in the four corners of the image.

If we allow normalization with respect to scaling, too, the situation changes. Since the velocity grows quadratically in the field but linearly in the pattern, the two do not really match. The similarity will become higher for smaller patches of the field. Because there, the difference in growth is less apparent. The positions of the best matches in Figure 8 bottom are pretty much as before in Figure 7 and Figure 8 top left. But the scale is much smaller and the similarity much higher (46 and 59). The new radius is 0.2 which is the minimal scale we looked for.

A very surprising output is produced, if the normalization is performed with respect to translation, rotation, and scaling on the left of Figure 9. The invariance with respect to background flow makes the whole field similar to a saddle and the best matches with a similarity of 64 do not coincide with the visible positions of the saddles. To illustrate the reason for this situation, we removed the background flow in the area of the highest similarity on one side. The underlying saddle indeed is less bent than the visible one. On the right in Figure 9, we can see the output of the algorithm, if a minimal scale of 0.1 is chosen. This area is small enough to return the expected

Minimal scale is 0.2. The back-ground flow has been removed in the right area.
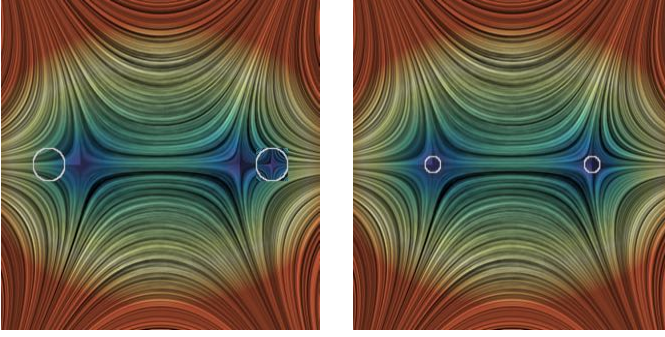
Minimal scale is 0.1

Fig. 9. Output of the algorithm, if normalization is performed w. r. t translation, rotation, and scaling.

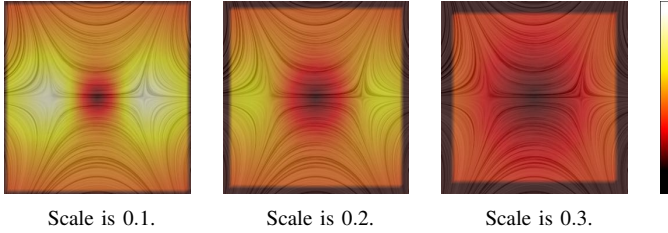match without being distorted too much. The similarity here is 94.

Scale is 0.1.   Scale is 0.2.   Scale is 0.3.

Fig. 10. Similarity of the saddle to the field with different scales. The similarity at every position is encoded in the color map.

In order to better explain the influence of the minimal scale to the results in Figure 9, we omitted the calculation of the maxima and the circles, but encoded the immediate similarity at every position for a given scale in the color map in Figure 10. The moments are normalized with respect to translation, rotation, and scaling. We can see that the image resembles a saddle pretty much everywhere except for the center. The higher the scale, the more of this central area is enclosed in the patches around the visible centers of the saddles, and the less similar becomes the result.

Finally we look for a vortex

$$f(z) = i\bar{z} \tag{32}$$

in a field of the same vortex but with opposite flow direction

$$f(z) = -i\bar{z}. \tag{33}$$

Since the vortices can not be mapped to each other using translation, rotation, or scaling, but only by a reflection, no significant similarity (0.3) can be found using only the three mappings. There is no visible output. Therefore, the pattern, the field, and the output of the algorithm can all be seen in Figure 11 left, because the line integral convolution does not show the orientation of the flow. If we normalize with respect to reflection, the algorithm returns the equality of the pattern and the field, compare Figure 11 right.

Pattern (32), field (33), and "output" without reflection
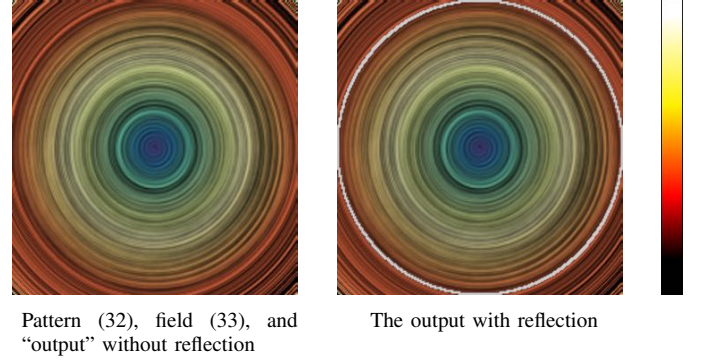
The output with reflection

Fig. 11. Illustration of the influence of normalization with respect to reflection. The color bar shows the similarity.

The preceding experiments lead to the following conclusion. The normalization is a powerful tool and the transformations have to be chosen carefully and wisely to fulfill their purpose in the particular applications. Especially the normalization with respect to background flow will sometimes lead to unexpected results. Because it gives the maximal similarity of the field in any constantly moving frame.

8 EXPERIMENTS

Our algorithm is based on the standard complex moments. We only changed the computation of the invariants. That means the numerical behavior is equal to the results given by Abu-Mostafa and Psaltis [1] and Teh and Chin [29] for complex moments. Our practical experiments support their fundamental findings.

The theory states full invariance for our moments, see Section 4. But in practical applications, this is not the case due to discretization errors. To investigate the practical reliability, we performed some experiments with discretized data for the saddle $v(x) = (x_2, x_1)^T = \bar{z} = f(z)$ on a uniform Cartesian grid $x = j/n, y = j/n, j = 1, 2, ...n$. The complex moments up to a given grade span a feature space. The error is measured as the Euclidean distance in this vector space. We show results of these experiments depending on the integration step size $1/n$ and on the maximum grade of the moments in Figure 12.
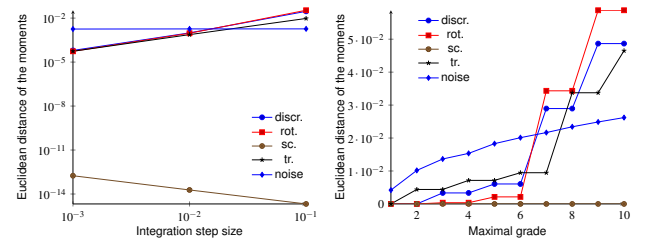

Fig. 12. Errors due to discretization with a resolution of 0.1 (discr.), total rotation (rot.), outer translation (tr.), outer scaling (sc.), and evenly distributed noise with $SNR = 3.5$ (noise). The lines connecting the points are for visualization purposes only.

First, we compared the calculated moments to the analytic values. The corresponding graphs in Figure 12 are marked

by "discr.". The error depends linearly on the resolution but grows faster with increasing grade. The 'jumps' after every increment by two is due to the structure of the moments. The saddle is only represented by moments of odd grade.

To analyze the invariance of the moments, we rotated and scaled the saddle and added uniform background flow with different directions and velocity. Figure 12 shows the largest differences of the moment invariants for these transformations. The corresponding graphs are marked by "rot.", "sc.", and "tr.". As can be seen, the errors with respect to rotation and translation are in the order of the resolution of the discretization. Only the invariance with respect to scaling is close to perfect. Since the background flow is only represented by moments of even grade, the 'jumps' after every second increment of translation is shifted compared to the 'jumps' that are linked with the saddle.

Finally, we tested robustness with respect to evenly distributed noise. The resulting error for a signal to noise ratio of $SNR = 3.5$ is shown in the graphs marked by "noise" in Figure 12. The influence of the noise scales linearly with respect to its power. The behavior of the moments with respect to the chosen noise intensity is also representative for other noise magnitudes.

9 APPLICATIONS

The original flow field, in which we look for the patterns.
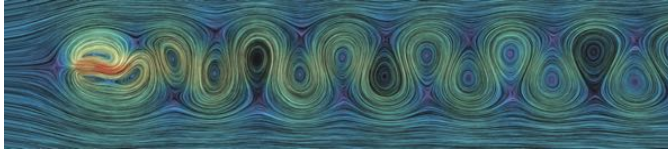
The field with removed mean flow serves as basis for the pattern selection.

Fig. 13. Line integral convolution of the dataset. The colors represent the velocity of the field: blue is slow, red is fast.

We applied our algorithm to one time slice of a 2D CFD simulation of a Kármán vortex street, which is the result of a flow passing a cylinder. The line integral convolution (LIC) [4] of this slice can be found in Figure 13. We calculated the complex moments for a discrete number of positions and scales in the field to cover the inner translation and scaling invariance. Then, we normalized the moments according to Corollary 1. As similarity measure, we used the reciprocal of the minimum of the Euclidean distances of the set of moment invariants up to a given grade.

In the following examples, we select query patterns from the dataset field without mean flow (Figure 13 bottom) and search for it in the original dataset (Figure 13 top). The chosen features are shown in Figure 14. Results of our algorithm for a maximal grade of three and five applied to the vortex saddle

Vortex saddle pattern in Figures 15, 16, and 17

Double vortex pattern in Figure 18

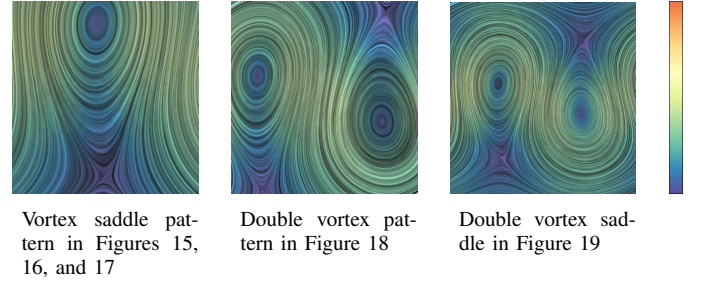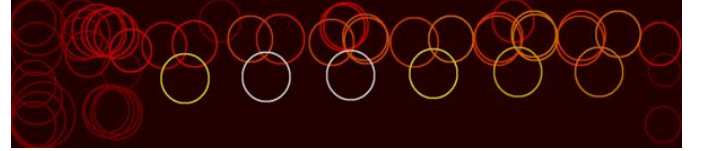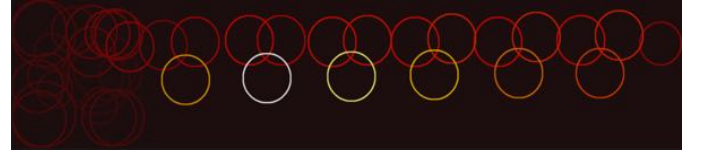Double vortex saddle in Figure 19

Fig. 14. Query patterns selected from Figure 13 bottom.

The moments up to a maximal grade of three were used.

The moments up to a maximal grade of five were used.

Fig. 15. Similarity of the dataset to the vortex saddle pattern.

combination on the left of Figure 14 can be found in Figure 15. It confirms the invariance with respect to outer translation, the similarity field takes its maximum at exactly the position and the size, where the pattern itself was selected.

To show that our algorithm works adequately, we overlay its output for the saddle vortex combination from Figure 15 over the LIC of the field without the mean flow (Figure 13 bottom). Figure 16 shows the result, which allows some interesting observations:

- As expected, the maximum similarity appears where the pattern meets itself and the following local maxima appear where the pattern repeats itself along the Kármán street.
- At first sight, it might be surprising that there is no match at the saddle vortex combinations with the saddle in the upper half of the image. Even though the LIC image shows the same pattern, the flow orientation is reversed. Invariance with respect to reflection could be easily added to the set of transforms considered for normalization but we liked to keep the moments sensitive with respect to this feature to stress the difference in the vortices.
- There are matches with intermediate similarity on the upper left and right of each strong match. They highlight the rotated pattern by $\frac{2\pi}{3}$ and $\frac{4\pi}{3}$ that consist of the same vortex and one of the two upper saddles to the left and the right. The similarity is lower due to the slight, oval deformation of the vortices.
- A higher accumulation of approximately concentric circles and some apparently false positives can be observed

Fig. 16. For comparison, the similarity to the saddle vortex combination was laid over the LIC of the flow field with removed mean flow.

at the more distant repetitions. This phenomenon can be reduced by increasing the maximal grade of the moments, as shown in Figure 15 bottom. Here we used the 21 moments up to the fifth grade, which results in a higher discriminating power than using just the 10 moments up to third order.
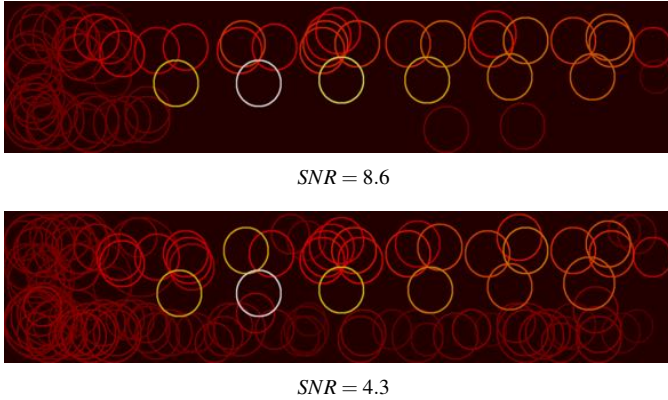

$SNR = 8.6$

$SNR = 4.3$

Fig. 17. Similarity of the dataset to the saddle vortex combination with distortions by different signal to noise ratios.

We analyzed the robustness of our algorithm with respect to noise. For this experiment we added a random field of evenly distributed noise to the data set. Some visual results can be found in Figure 17. Since the moments are computed by integration, they are very robust. The similarity values hardly change under the influence of small noise. The main change in the images is the many new circles with mostly rather low similarity. The reason for this, is not the calculation of the moment invariants but the decision to draw the circles at local maxima. The noise leads to a less smooth similarity field and therefore an increasing number of maxima. That is no disadvantage of the moment invariants because they are not intrinsically tied to the final visualization of the similarity field. The calculation of the similarity values starts to fail when the power of the noise gets close to twice that of the one of the image, which can be considered pretty robust.

The results of the algorithm in Figure 16 are quite representative. The mentioned observations can be made with other patterns, too. As another example, we show the output of our algorithm for the pattern consisting of the two counter

oriented vortices and two saddles from the right of Figure 14. Again, as expected, the original cutout can be found in the very bright circle and its repetitions with lower similarity along the Kármán street in Figure 19.

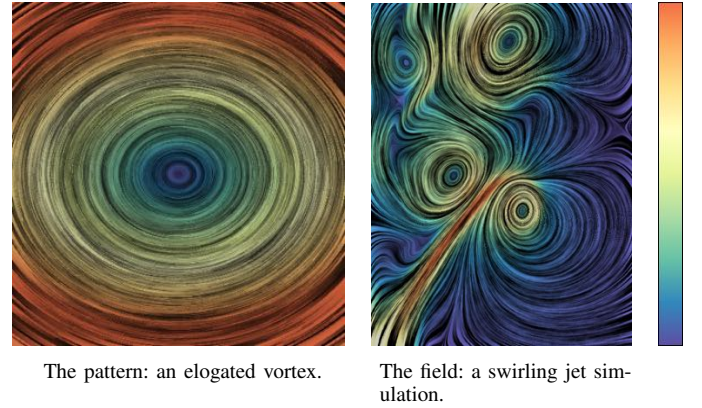The pattern: an elogated vortex.   The field: a swirling jet simulation.

Fig. 20. LIC color coded by velocity of the pattern and the field of our second application.

As another example, we applied our algorithm to the simulation of a swirling jet. Figure 20 show the pattern on the left and the field in the middle. The pattern is a slightly elongated vortex over the unit sphere defined by the formula

$$f(z) = -5iz + i\bar{z}. \qquad (34)$$

The flow field is the result of the simulation of a fast stream entering a liquid at rest. We show the output of our algorithm laid over the LIC in Figure 21. In the left image, we normalized with respect to rotation, reflection, translation, and scaling. The appearances that are in the vicinity of the vortices, but not in the center are due to the normalization with respect to translation. This effect was explained in Section 7. The image on the right is the result of normalization only with respect to rotation, reflection, and scaling. It is also apparent, that there are weak matches at bends with high curvature in the left image, which do no longer appear in the right one. Leaving out the background flow gives them the shape of vortices, too. If we leave out normalization with respect to reflections, only the vortices to the right of the jet are detected. Please note, that the maximal similarity values im the left image are much higher ($sim = 60$) than the ones without translational

Fig. 18. The similarity of the underlying field to the counter oriented double vortex is encoded in the brightness of the circles.

Fig. 19. For comparison, the similarity to the double vortex saddle pattern was laid over the LIC of the field with removed mean flow.
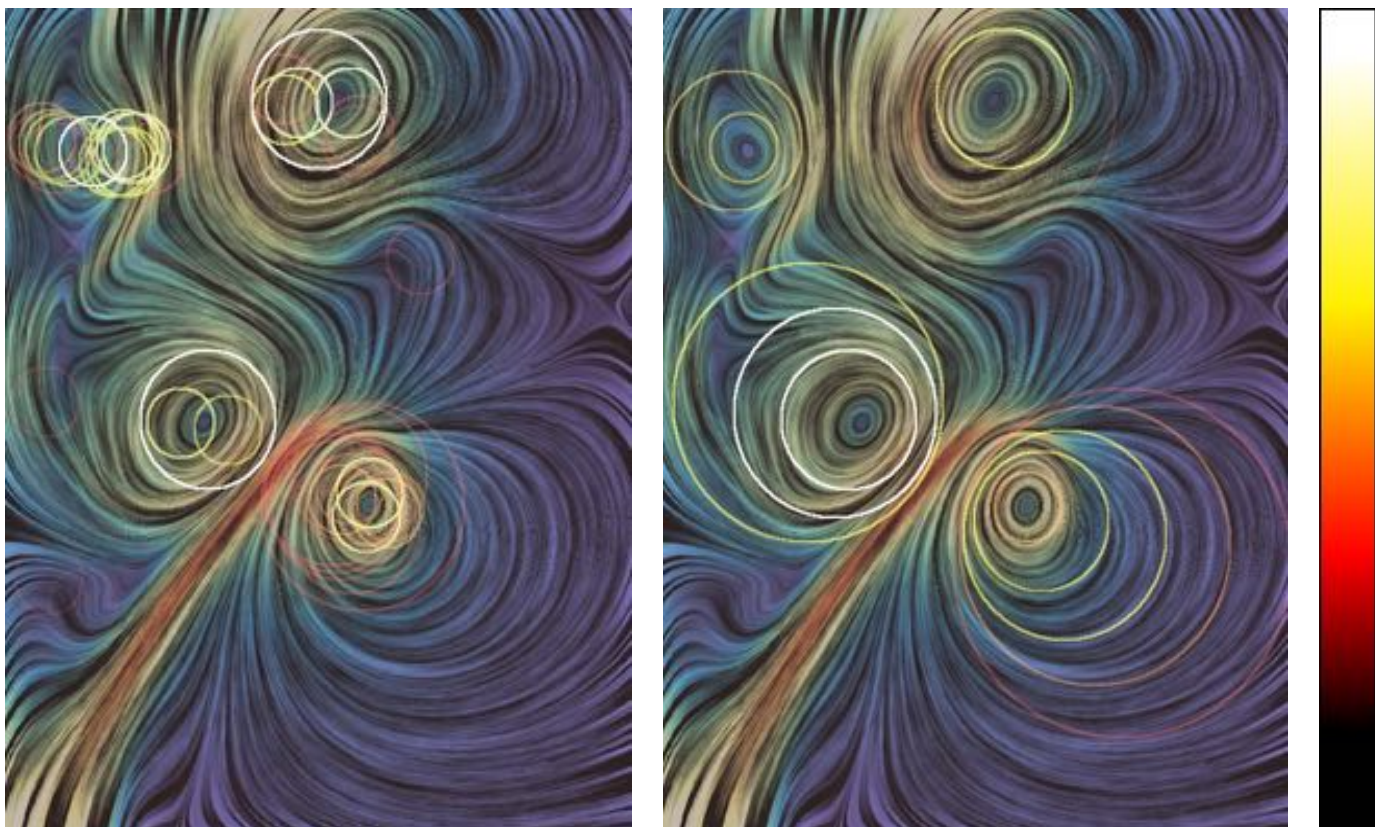
Normalized with respect to rotation, reflection, scaling, and translation.

Normalized with respect to rotation, reflection, and scaling.

Fig. 21. For comparison, the similarity to the elongated vortex pattern from (34) was laid over the LIC of the simulated field from Figure 20. The colorbar indicates the similarity.

normalization ($sim = 7$). We adapted the range of the colormap to make both results visible.

10 CONCLUSION

In this paper, we have introduced moment normalization for vector fields to define a new class of moment invariants as descriptors for vector fields. We have presented the theoretical framework for the calculation of moment invariants of 2D flow fields using this technique. By applying it to a real world data set, we could show that the mathematical results can be used more generally to describe, analyze, and compare discrete flows in a numerically robust way.

Compared to the invariants suggested by Schlemmer et al., our approach exhibits a couple of advantages. It is intuitively motivated and produces a complete and independent set of moment invariants, can easily be generalized to other transformations, as for example reflections, and to other function space bases. It considers the velocity of the field and thus overcomes the problem of self similarity of vortex like structures revealing the size of the patterns as maxima in the scale space. The order of the moments is not limited, resulting in a substantially higher discriminative power.

The complexity for the computation of the moments is the same as in the work of Schlemmer et al. The computation of the moments corresponds to a convolution and can be efficiently implemented using the FFT. Our current implementation does not focus on an optimal performance. For our examples the runtime is approximately 1 minute using the FFT, which we still consider feasible. But when moving to 3D the runtime becomes a challenge.

In our future work, we plan to extend the moment normalization to time dependent and 3D flow fields. The generalization to 3D involves a couple of challenges, but is possible.

ACKNOWLEDGMENTS

REFERENCES

[1] Y. S. Abu-Mostafa and D. Psaltis. Recognitive Aspects of Moment Invariants. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 698–706, 1984.
[2] A. Brambilla, R. Carnecky, R. Peikert, I. Viola, and H. Hauser. Illustrative Flow Visualization: State of the Art, Trends and Challenges. *EG 2012*, State of the Art Reports:75–94, 2012.
[3] R. Bujack, I. Hotz, G. Scheuermann, and E. Hitzer. Moment Invariants for 2D Flow Fields via Normalization. In *IEEE Pacific Visualization Symposium, PacificVis 2014 in Yokohama, Japan*, 2014.
[4] 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*, SIGGRAPH '93, pages 263–270. ACM, 1993.
[5] J. Daniels, E. W. Anderson, L. G. Nonato, and C. T. Silva. Interactive Vector Field Feature Identification. *IEEE Trans. on Vis. and Computer Graphics*, 16(6):1560–1569, 2010.
[6] H. Dirilten and T. Newman. Pattern Matching Under Affine Transformations. *IEEE Trans. on Computers*, 26(3):314–317, 1977.
[7] J. Ebling. *Visualization and Analysis of Flow Fields using Clifford Convolution*. PhD thesis, University of Leipzig, Germany, 2006.
[8] J. Ebling and G. Scheuermann. Pattern Matching on Vector Fields using Gabor Filter. In *Proceeding from Visualization, Imaging, and Image Processing (VIIP)*, 2004.
[9] G. Erlebacher, C. Garth, R. S. Laramee, H. Theisel, X. Tricoche, T. Weinkauf, and D. Weiskopf. Texture and Feature-Based Flow Visualization - Methodology and Application. In *IEEE Visualization Tutorial*, 2006.
[10] J. Flusser. On the independence of rotation moment invariants. *Pattern Recognition*, 33(9):1405–1410, 2000.
[11] J. Flusser, B. Zitova, and T. Suk. *Moments and Moment Invariants in Pattern Recognition*. Wiley, 2009.
[12] E. Heiberg, T. Ebbers, L. Wigström, and M. Karlsson. Three-Dimensional Flow Characterization Using Vector Pattern Matching. *IEEE Trans. on Vis. and Computer Graphics*, 9(3):313–319, 2003.
[13] M.-K. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187, 1962.
[14] J. Jeong and F. Hussain. On the Identification of a Vortex. *Journal of Fluid Mechanics*, 285:69–94, 1995.
[15] J. Kasten, I. Hotz, and H.-C. Hege. On the Elusive Concept of Lagrangian Coherent Structures. In *Topological Methods in Data Analysis and Visualization II. Theory, Algorithms, and Applications. (TopoInVis'11)*, pages 207–220, 2012.
[16] J. Kasten, J. Reininghaus, I. Hotz, and H.-C. Hege. 2D Time-dependent Vortex Regions based on the Acceleration Magnitude. *IEEE Trans. on Vis. and Computer Graphics*, 17(12):2080–2087, 2011.
[17] R. S. Laramee, H. Hauser, L. Zhao, and F. H. Post. Topology-Based Flow Visualization, The State of the Art. In *Topology-based Methods in Visualization*, pages 1–19, 2007.
[18] W. Liu and 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)*, pages 522–531, 2010.
[19] T. McLoughlin, R. S. Laramee, R. Peikert, F. H. Post, and M. Chen. Over Two Decades of Integration-Based, Geometric Flow Visualization. In *EG 2009 - State of the Art Reports*, pages 73–92, 2009.
[20] C. E. Moxey, S. J. Sangwine, and T. A. Ell. Hypercomplex Correlation Techniques for Vector Images. *Signal Processing, IEEE Transactions*, 51(7):1941–1953, 2003.
[21] F. Sadlo, R. Peikert, and E. Parkinson. Vorticity Based Flow Analysis and Visualization for Pelton Turbine Design Optimization. In *Proceedings of the Conference on Visualization '04*, pages 179–186, 2004.
[22] J. Sahner, T. Weinkauf, N. Teuber, and H.-C. Hege. Vortex and Strain Skeletons in Eulerian and Lagrangian Frames. *IEEE Trans. on Vis. and Computer Graphics*, 13(5):980–990, 2007.
[23] T. Salzbrunn, H. Jänicke, T. Wischgoll, and G. Scheuermann. The State of the Art in Flow Visualization: Partition-based Techniques. In *Simulation and Visualization 2008 Proceedings*, 2008.
[24] M. Schlemmer. *Pattern Recognition for Feature Based and Comparative Visualization*. PhD thesis, Germany, 2011.
[25] M. Schlemmer, M. Heringer, F. Morr, I. Hotz, M.-H. Bertram, C. Garth, W. Kollmann, B. Hamann, and H. Hagen. Moment Invariants for the Analysis of 2D Flow Fields. *IEEE Trans. on Vis. and Computer Graphics*, 13(6):1743–1750, 2007.
[26] M. Schlemmer, I. Hotz, B. Hamann, and H. Hagen. Comparative Visualization of Two-Dimensional Flow Data Using Moment Invariants. In *Proceedings of Vision, Modeling, and Visualization (VMV'09)*, volume 1, pages 255–264, 2009.
[27] D. Schneider, A. Wiebel, H. Carr, M. Hlawitschka, and G. Scheuermann. Interactive Comparison of Scalar Fields Based on Largest Contours with Applications to Flow Visualization. *IEEE Trans. on Vis. and Computer Graphics*, 14(6):1475–1482, 2008.
[28] M. R. Teague. Image analysis via the general theory of moments*. *J. Opt. Soc. Am.*, 70(8):920–930, 1980.
[29] C.-H. Teh and R. Chin. On image analysis by the methods of moments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(4):496–513, 1988.

Roxana Bujack received the Diploma in mathematics in 2010 and the BSc in computer science in 2011 at Leipzig University in Germany. She is currently a PhD student in the Image and Signal Processing group of the Department of Computer Science at Leipzig University. Her research interests include moment invariants, flow visualization, pattern recognition in vector fields and Clifford analysis.

Ingrid Hotz received the M.S. degree in theoretical Physics from the Ludwig Maximilian University in Munich Germany and the PhD degree from the Computer Science Department at the University of Kaiserslautern, Germany. During 2003 – 2006 she worked as a postdoctoral researcher at the Institute for Data Analysis and Visualization (IDAV) at the University of California. From 2006 – 2013 she was the leader of a junior research group at the Zuse Institute in Berlin Germany. Currently, she is a scientific visualization group leader at the German Aerospace Center (DLR). Her research interests are in the area of data analysis and scientific visualization with focus on tensor and vector fields.

Gerik Scheuermann received the master degree (diplom) in mathematics in 1995 and a PhD degree in computer science in 1999, both from the Technical University of Kaiserslautern. He is a full professor at the University of Leipzig since 2004. He is coauthor of more than 170 reviewed book chapters, journal, and conference papers. His current research interests focus on visualization, especially on topology-based methods, flow visualization, document visulaization and visualization for life sciences. He has served as paper cochair for Eurovis 2008, IEEE SciVis 2011, IEEE 2012, and on various program committees including IEEE Visualization, Eurovis, and IEEE PacificVis. He has organized TopoInVis 2007, AGACSE 2008 and EuroVis 2013.

Eckhard Hitzer is Senior Associate Professor at the Department of Material Science at the International Christian University in Mitaka/Tokyo, Japan. His special interests are theoretical and applied Clifford geometric algebras and Clifford analysis, including applications to crystal symmetry visualization, neural networks, signal and image processing. Additionally he is interested in environmental radiation measurements, renewable energy and energy efficient buildings.