

# Moment Invariants for Multi-Dimensional Data

Roxana Bujack and Hans Hagen

**Abstract** Moment invariants have long been successfully used for pattern matching in scalar fields. By their means, features can be detected in a data set independent of their exact orientation, position, and scale. Their recent extension to vector fields was the first step towards rotation invariant pattern detection in multi-dimensional data.

In this paper, we propose an algorithm that extends the normalization approach to tensor fields of arbitrary rank in two and three dimensions.

## 1 Introduction

Tensor fields play an important role in the study of many physical phenomena. Earthquakes, volcanoes, diffusion, or deformation can all be described using tensor fields. Higher derivatives of scalar and vector fields also form tensors. In contrast to their lower rank counterparts, tools for the analysis of higher rank tensor fields are not as well developed. In particular, the visualization of three-dimensional tensor fields can suffer from a clutter. Any given visualization element may occlude elements behind it. The question of what is included or omitted in a visualization is very important and can potentially impact the scientific understanding. Pattern detection can help to address this problem by reducing the areas that are drawn to the locations of features that are of importance to the analyst. In this paper, we suggest an algorithm for rotation invariant pattern detection for tensor fields of arbitrary rank.

---

Roxana Bujack  
Technical University Kaiserslautern, e-mail: bujack@cs.uni-kl.de

Hans Hagen  
Technical University Kaiserslautern e-mail: hagen@cs.uni-kl.de

Rotation invariance is a critical requirement of pattern detection to minimize the set of unique patterns required in a search. Moment invariants allow one to achieve rotation invariance without the need for point to point correlations.

Moments are the projections of a function with respect to a function space basis. We can think of them as the coordinates that represent the pattern. They can then be used to construct moment invariants - values that do not change under certain transformations. In this paper, we concentrate on orthogonal transformation and isotropic scaling.

There are two main approaches to the construction of moment invariants: normalization and the definition of a generator [17]. For *normalization*, a standard position is defined by demanding certain moments to assume fixed values and all functions are transformed to match it. Then the remaining moments form a complete and independent set of moment invariants. The *generator* approach relies on defining an explicit rule on how to combine the moments in a way that suppresses the alignment information, usually by multiplication and addition.

The main disadvantage of the normalization approach is that it is unstable if the moment chosen for normalization should become zero. On the other hand, the main disadvantage of the generator approach is the difficulty in finding and proving the existence of an independent moment. Depending on the application, one method may be more effective than the other.

Both approaches have been applied to generate moment invariants for scalar fields and recently also to vector fields. To the authors' best knowledge, to date, moment invariants for matrix fields or higher rank tensor fields have only been presented using a generator approach [25].

In this paper, we present an algorithm that constructs moment invariants using normalization for two- and three-dimensional tensor fields of arbitrary rank. For scalar fields, the zeroth and first order moments are usually used for the normalization with respect to translation and scaling. This is why the standard position with respect to orientation is generally chosen to be the Jordan normal form of the second order moments, which is related to the principal axes of the covariance matrix. For vector fields, Bujack et al. [5] use the Schur form of the first order moments. This is also of second rank. For higher rank tensor fields, the first order moments are already of a rank higher than two. Hence, matrix algebra approaches can no longer be applied.

Our solution to this problem is to use tensor contractions that produce first rank tensors from higher rank moment tensors. These first rank tensors behave like vectors and can easily assume a standard position. A similar approach has been used in [10] to generate a normalizer for 2D affine transformations and in [27, 9] for 3D scalar functions. We extend this idea to generate rotation invariants for tensor valued functions.

The main contributions of this paper are as follows:

- We propose a methodology to apply tensor algebra to the normalization of moments of tensor fields of any order  $o \in \mathbb{N}$ , including scalar, vector, and matrix fields.

- By producing a complete and independent set of moment invariants, this method can provide a solution for tensor fields where the generator approach fails [25].
- The flexibility inherent in this method also improves on state the art normalization approaches for vector and scalar data by finding solutions where the current techniques fail due to vanishing moments.
- To our knowledge, this is the first time that moment invariants have been computed for a tensor field of second order.

## 2 Related Work

The first moment invariants were introduced to the image processing society by Hu [21]. The development of moment tensors by Diriltan and Newman [11] extended moment invariants to three-dimensional data. Pinjo, Cyganski, and Orr [27] calculated 3D orientation estimation from moment contraction to first order moments.

Please note that invariants can be constructed not only from moments, but, for example, from derivatives, too [12, 13]. The fundamental theorem of moment invariants [28] guarantees that every algebraic invariant has a moment invariant counterpart.

In his seminal work [14], Flusser presented a calculation rule to generate a complete and independent set for 2D scalar functions. Later, he proved that it further solves the inverse problem in [15].

For three-dimensional functions, the task is much more challenging. One research path goes in the direction of the spherical harmonics. They are an irreducible representation of the rotation group and therefore an adequate basis for the generation of moment invariants. Lo and Don [26], Burel and Henocq [6], Kazhdan et al. [22], Canterakis [8], and Suk et al. [33], use them to construct moment invariants for three-dimensional scalar functions.

A second research path makes use of the tensor contraction method, as described by Diriltan and Newman [11]. While all tensor contractions to zeroth rank are rotationally invariant, it can be difficult to find a complete and independent set. Suk and Flusser propose to calculate all possible zeroth rank contractions from moment tensors up to a given order and then skip the linearly dependent ones in [32]. Higher order dependencies still remain.

Schlemmer et al. [30, 31] generalized the notion of moment invariants to vector fields. Later Bujack et al. [3] provided a normalization method that is flexible with respect to the choice of the normalizer. It leads to a complete set of independent moment invariants for 2D flow fields that is applicable to any pattern. Their extension to three-dimensional vector fields uses the transformation properties of the second rank tensors and their eigenvectors to define a standard position [5]. This method requires the second rank tensor to not vanish.

Langbein and Hagen [25] treat tensor fields of higher rank. They show that the tensor contraction method can be generalized to arbitrary tensor fields. However,

their suggested method to reduce the redundancy of the generator approach uses exact calculation. Its application in a discrete setting (i.e., programmatically) has not yet been practically applied. In this paper, we introduce a normalization method for tensor fields which automatically results in an independent and complete set. We demonstrate its utility by applying it to pattern detection in analytic as well as simulation data.

The state of the art of moment invariants with respect to the each of the two approaches, the different data types and dimensions are summarized in Tables 1 and 2. The attributes have the following meaning:

- *Complete*: The set is complete if any arbitrary moment invariant can be constructed from it.
- *Independent*: The set is independent if none of its elements can be constructed from its other elements.
- *Flexible*: The set is flexible w.r.t. vanishing moments if it exists for any pattern, meaning it does not rely on any specific moment  $c_{p_0,q_0}$  to be non-zero.

Dim.	Data type	Authors	Complete	Independent	Flexible
2D	Scalar	Flusser et al. [17]	✓	✓	-
2D	Vector	Bujack et al. [3]	✓	✓	✓
2D	Tensor	-	-	-	-
3D	Scalar	Cyganski et al. [9]	✓	✓	-
3D	Vector	Bujack et al. [5]	✓	✓	-
3D	Tensor	-	-	-	-

Table 1: State of the art of moment invariants constructed from normalization.

Dim.	Data type	Authors	Complete	Independent	Flexible
2D	Scalar	Flusser et al. [14]	✓	✓	-
2D	Vector	Schlemmer et al. [29]	-	-	-
2D	Tensor	Langbein et al. [25]	?	(✓)	-
3D	Scalar	Flusser et al. [32]	?	-	?
3D	Vector	Langbein et al. [25]	?	(✓)	-
3D	Tensor	Langbein et al. [25]	?	(✓)	-

Table 2: State of the art of moment invariants constructed using the generator approach. The brackets indicate that independence is given only theoretically and the question mark that this property is unknown.

### 3 Theory

We will start by reviewing the theoretical underpinnings of our algorithm so that readers from both the visualization world and the mathematical world start from common ground. We also show how to generate tensors that transform like vectors under rotations and reflections.

#### 3.1 Tensors and Transformations

Tensors are a natural representation of physical quantities that follow specific rules under transformations of the coordinate system. They can be represented as arrays of numbers relative to a fiducial basis. The rank of a tensor corresponds to the number of indices that we need to identify the different numbers in the array. Scalars are tensors of rank zero, vectors are tensors of rank one, and matrices are tensors of rank two. The interested reader can find exemplary introductions to tensor analysis, in addition to the definitions and lemmata we review in this work in [1] or in [18]. We will make use of the Einstein notation where the summation symbol is dropped in products over the same index. The summation is performed from 1 to  $d$ , in which the latter is the underlying dimension. Please note that the theory is valid for arbitrary  $d \in \mathbb{N}$ . Later, in our experiments, we will work with two- and three-dimensional fields.

**Definition 1** A multidimensional array  $T_{j_1 \dots j_m}^{i_1 \dots i_n}$  that, under an active transformation by the invertible matrix  $A_j^i \in \mathbb{R}^{d \times d}$ , behaves as:

$$T_{j_1 \dots j_m}^{i_1 \dots i_n} = |\det(A^{-1})|^w A_{k_1}^{i_1} \dots A_{k_n}^{i_n} (A^{-1})_{j_1}^{l_1} \dots (A^{-1})_{j_m}^{l_m} T_{l_1 \dots l_m}^{k_1 \dots k_n}, \quad (1)$$

is called a (relative, axial) **tensor** of covariant rank  $m$ , contravariant rank  $n$ , and weight  $w$ . An (absolute) tensor has weight zero.

**Remark 1** An active transformation,  $x' = Ax$ , transforms the field, but not the frame. Thus the coordinate system remains unchanged and a tensor field  $T : \mathbb{R}^d \rightarrow \mathbb{R}^{d^n \times d^m}$  is transformed via  $T'(x') = T(x)$ . One can, for example, rotate an object actively. This is in contrast to a passive rotation in which the coordinate system is rotated rather than the object itself.

**Example 1** A vector  $v \in \mathbb{R}^d$  is an absolute, contravariant, first rank tensor, because it behaves via  $v^i = \sum_{j=1}^d A_j^i v^j$  under active transformations. In Einstein notation, this is written as  $v^i = A_j^i v^j$ .

A matrix  $M \in \mathbb{R}^{d \times d}$  is a tensor of contravariant rank one and covariant rank 1, because it behaves via  $M^l = AMA^{-1}$  under active transformations. In Einstein notation, this is written as  $M^i_j = A_k^i (A^{-1})_j^k M_l^k$ .

**Remark 2** For orthogonal transformations  $A \in \mathbb{R}^{d \times d}$ , i.e. rotations and reflections, the distinction of the indices into covariant and contravariant ones from Definition 1 is not necessary because they satisfy  $A^T = A^{-1}$ . Further, the weight can be ignored, because they satisfy  $|\det A| = |\det(A^{-1})| = 1$ .

**Lemma 1** Let  $T$  and  $\tilde{T}$  be two relative tensors of covariant rank  $m$ , contravariant rank  $n$ , and weight  $w$  and  $\tilde{m}, \tilde{n}, \tilde{w}$  respectively. Then the product  $T \otimes \tilde{T}$  (also called outer product or tensor product):

$$(T \otimes \tilde{T})_{j_1 \dots j_m \tilde{j}_1 \dots \tilde{j}_{\tilde{m}}}^{i_1 \dots i_n \tilde{i}_1 \dots \tilde{i}_{\tilde{n}}} := T_{j_1 \dots j_m}^{i_1 \dots i_n} \tilde{T}_{\tilde{j}_1 \dots \tilde{j}_{\tilde{m}}}^{\tilde{i}_1 \dots \tilde{i}_{\tilde{n}}} \quad (2)$$

is a relative tensor of covariant rank  $m + \tilde{m}$ , contravariant rank  $n + \tilde{n}$ , and weight  $w + \tilde{w}$ .

**Lemma 2** Let  $T$  be a relative tensor of covariant rank  $m$ , contravariant rank  $n$ , and weight  $w$ . Then the contraction  $\sum_{(i_k, j_l)} T$  of a covariant index  $i_k$  and a contravariant index  $j_l$

$$\left( \sum_{(i_k, j_l)} T \right)_{j_1 \dots j_{l-1} j_{l+1} \dots j_m}^{i_1 \dots i_{k-1} i_{k+1} \dots i_n} := T_{j_1 \dots j_{l-1} j_{l+1} \dots j_m}^{i_1 \dots i_{k-1} i_{k+1} \dots i_n} \quad (3)$$

is a relative tensor of covariant rank  $m - 1$ , contravariant rank  $n - 1$ , and weight  $w$ .

**Remark 3** Please note that  $\sum_{(i_k, j_l)}$  is the symbol for the contraction of the two indices  $i_k$  and  $j_l$  as used in [25]. It is different from the sum  $\sum_{i_k, j_l}$  over these indices. In particular, the contracted indices are no longer indices of the contracted tensor.

**Example 2** The product of a matrix  $M \in \mathbb{R}^{d \times d}$  and a vector  $v \in \mathbb{R}^d$  is a tensor product  $\otimes$  followed by a contraction  $\sum_{(2,1)}$  of the second covariant with the first contravariant index

$$\left( \sum_{(2,1)} (M \otimes v) \right)^i = \left( \sum_{(2,1)} (M_j^i v^j) \right)^i = M_j^i v^j. \quad (4)$$

Also, the trace of a matrix  $M$  can be written as the contraction by  $M_i^i$ .

### 3.2 Moment Tensors

Dirilten and Newman suggest the use of moment tensors for the construction of moment invariants with respect to orthogonal transforms in [11]. They construct the moment tensors by arranging the moments of each order in a way such that they obey the tensor transformation property (1).

**Definition 2** For a scalar function  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  with compact support, the **moment tensor**  ${}^o M$  of order  $o \in \mathbb{N}$  takes the shape

$${}^oM^{k_1 \dots k_o} = \int_{\mathbb{R}^d} x^{k_1} \dots x^{k_o} f(x) \mathrm{d}^d x, \quad (5)$$

with  $l \in \{1, \dots, o\}$ ,  $k_l \in \{1, \dots, d\}$ , and  $x^{k_l}$  representing the  $k_l$ -th component of  $x \in \mathbb{R}^d$ .

This arrangement of the moments of the same order into arrays simplifies the calculation of their behavior under linear transformations, which is very helpful for the construction of moment invariants. Cyganski et al. [10] use moment tensors to determine the orientation of scalar functions and to normalize with respect to linear transformations. In [9], they present the following important theorem.

**Theorem 1** *The moment tensor of order  $o$  of a scalar function is a contravariant tensor of rank  $o$  and weight  $-1$ .*

Langbein et al. [25] have generalized the definition of the moment tensor to tensor valued functions.

**Definition 3** *For a tensor field  $T : \mathbb{R}^d \rightarrow \mathbb{R}^{d^n \times d^n}$  with compact support, the **moment tensor**  ${}^oM$  of order  $o \in \mathbb{N}$  takes the shape*

$${}^oM = \int_{\mathbb{R}^d} x^{\otimes o} \otimes T(x) \mathrm{d}^d x, \quad (6)$$

where  $x^{\otimes o}$  denotes the  $o$ -th tensor power of the vector  $x$ .

The analogy between Definitions 2 and 3 can be seen more easily if we write Definition 3 using the indices

$${}^oM_{j_1 \dots j_m}^{i_1 \dots i_n k_1 \dots k_o} = \int_{\mathbb{R}^d} x^{k_1} \dots x^{k_o} T_{j_1 \dots j_m}^{i_1 \dots i_n}(x) \mathrm{d}^d x, \quad (7)$$

with  $l \in \{1, \dots, o\}$ ,  $k_l \in \{1, \dots, d\}$ , and  $x^{k_l}$  representing the  $k_l$ -th component of  $x \in \mathbb{R}^d$ .

The following theorem is the main theoretical contribution of this paper. It allows the construction of moment invariants for tensor fields analogously to the ones for scalar fields.

**Theorem 2** *The moment tensor of order  $o$  of a tensor field of covariant rank  $m$ , contravariant rank  $n$  and weight  $w$  is a tensor of covariant rank  $m$ , contravariant rank  $n + o$  and weight  $w - 1$ .*

*Proof.* The vector  $x$  is an absolute, contravariant, first rank tensor. Application of Lemma 1 shows that  $x^{\otimes o} \otimes T(x)$  is a tensor of covariant rank  $m$ , contravariant rank  $n + o$  and weight  $w$ .

The decrease of the weight by one comes from the integral, because under a change of the integration variable  $x' = Ax$ , the infinitesimal element is multiplied by the functional determinant  $\mathrm{d}^d x' = |\det A| \mathrm{d}^d x$ .

The following Corollary is the property used by Langbein and Hagen [25] to construct moment invariants. They generate the invariants from moment tensor contractions to scalars.

**Corollary 1** *The rank zero contractions of any product of the moment tensors are moment invariants with respect to rotation and reflection.*

*Proof.* It follows from Lemma 1 that any combination of moment tensor factors in a tensor product is a tensor. Contraction of this product to zeroth rank is a tensor because of Lemma 2. According to Definition 1, this zeroth rank tensor satisfies  ${}^0M' = {}^0M$  because rotations and reflections satisfy  $|\det A| = 1$ .

Analogously to the previous corollary, we construct tensors of first and second rank from contractions and products of moment tensors.

**Corollary 2** *The rank one contractions of any product of the moment tensors behave like vectors under rotation and reflection.*

*Proof.* Lemmata 1 and 2 guarantee that the first order contractions are tensors. According to Definition 1, it satisfies

$${}^1M'^i = A_j^{i1} M^j, \quad (8)$$

because rotations and reflections satisfy  $|\det A| = 1$ . This corresponds to the classical matrix vector product  ${}^1M' = A^1 M$ .

**Corollary 3** *The rank two contractions of any product of the moment tensors behave like matrices under rotation and reflection.*

*Proof.* Lemmata 1 and 2 guarantee that the rank one contractions are tensors. According to Definition 1, it satisfies

$${}^2M'^{i_1 i_2} = A_{j_1}^{i_1} A_{j_2}^{i_2} M^{j_1 j_2}, \quad (9)$$

because rotations and reflections satisfy  $|\det A| = 1$ . Because they also satisfy  $A^T = A^{-1}$ . This coincides with the matrix product  ${}^2M' = A^2 M A^{-1}$ .

## 4 Algorithm and Complexity

In the two-dimensional case, a rotation has one degree of freedom. That means a standard position can be defined using one vector, for example, by demanding this vector to align with the positive real axis.

In the three-dimensional case, we need two vectors to normalize w.r.t. a rotation. As a standard position, we choose the first one to align with the positive x-axis and the second one to lie in the upper half of the x-y-plane, which corresponds to a positive y-coordinate, compare Figure 1. For the remainder of the paper, we will describe the three-dimensional situation keeping in mind that the second vector will not be needed in 2D.

Theoretically, it does not matter which first rank contractions we choose as long as they are the same for the normalization of the pattern and the field. In the case

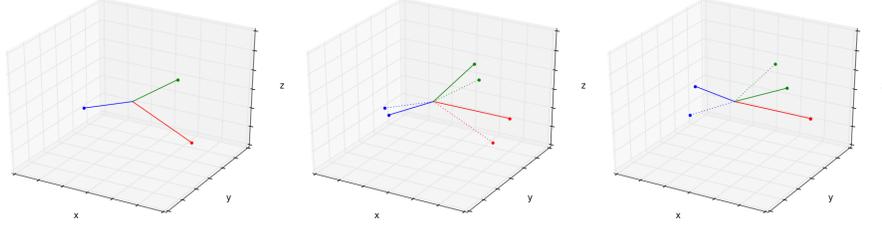


Fig. 1: Illustration on the 3D normalization wrt. rotation. The three vectors represent different normalizer candidates, i.e. first order moment tensor contractions. To maximize robustness, we use the vector with the highest magnitude (red) and align it with the positive x-axis. Since, the green vector has the highest magnitude orthogonal to the red one, it is used for the second step and rotated into the x-y-plane.

of a scalar field, for example, we could use either the first order moment tensor  ${}^1M$  or a contraction of the third order moment tensor  $\sum_{(1,2)} {}^3M$  to normalize with respect to rotation. However, for the stability of the algorithm, it is important that the tensors chosen for the normalization are not close to zero nor are linearly dependent. In our scalar field example, let us assume that the pattern of interest has no linear component and  ${}^1M = 0$ . It is impossible to rotate the zero vector onto the x-axis, which is why it is better to use the contraction  $\sum_{(1,2)} {}^3M \neq 0$  for the normalization instead. Even if the tensor is not exactly zero, but numerically small, using it impairs the robustness. Its orientation may change significantly if the field gets disturbed by only small noise, which would lead to a different standard position.

In practice, the moments are calculated up to a maximal given order  $o$ , which results in tensors  $\{T\}$  up to rank  $o+n+m$ . To maximize robustness, we construct all possible combinations of tensor products  $\{\tilde{T}\}$  not exceeding a given upper rank of  $o' > o$  from the tensors  $\{T\}$ . Then, we calculate all first rank contractions  $\{{}^1\tilde{T}\}$  from this set of products  $\{\tilde{T}\}$ . We allow the user to set the orders  $o$  and  $o'$ . We choose the vector with maximal magnitude  $v_1 := \operatorname{argmax}_{v \in \{{}^1\tilde{T}\}} \|v\|$  for the first normalization step. For the second step, we use the vector that is as orthogonal as possible to the first vector  $v_2 := \operatorname{argmax}_{v \in \{{}^1\tilde{T}\}} \|v_1 \times v\|$ .

Using only the first rank contractions has one important potential issue. If all odd ranked moment tensors are zero, we cannot construct any non-zero first rank contraction because the contraction always decreases the rank by two. To solve this problem, we make use of Corollary 3 and also generate all second rank tensors  $\{{}^2\tilde{T}\}$  from the products  $\{\tilde{T}\}$ . These behave like matrices and so, if they have eigenvectors, these are also possible vectors that could be used to assume the standard position. We restrict ourselves to the symmetric parts of the second rank tensors in order to guarantee the existence of real valued eigenvectors.

The stability of an eigenvector is determined by how distinguishable its corresponding eigenvalue is from the remaining eigenvalues. For example, a small perturbation of the matrix  $\operatorname{diag}(1, \varepsilon, -\varepsilon)$  with small  $\varepsilon$  could change the order of the two smaller eigenvalues. That changes the corresponding eigenvectors and the re-

sult of the normalization. On the other hand, the eigenvector that belongs to 1 is far more robust. Therefore, we weigh the vector with the minimal distance of its eigenvalue to the others. Let  $v_i$  be the eigenvectors that belong to the eigenvalues  $\lambda_i$  of the symmetric part of a second order contraction  $\{\tilde{T}^2\}$ . Then, we add the vectors  $\tilde{v}_i = v_i(\min_{j \neq i} |\lambda_i - \lambda_j|)$  to the set of first order contractions  $\{\tilde{T}^1\}$ . This increases the chances of finding a robust standard position. In our example, we would add  $(1 - \varepsilon, 0, 0)^T$ ,  $(0, 2\varepsilon, 0)^T$ , and  $(0, 0, 2\varepsilon)^T$ .

Please note that in contrast to real vectors, eigenvectors do not have a direction. For any eigenvector  $v$ ,  $-v$  is also an eigenvector. We therefore must keep track of two standard positions if an eigenvector is chosen as a normalizer. We do this by storing a second set of normalized moment tensors for the pattern.

In summary, the consecutive steps for the construction of a complete and independent set of moment invariants for tensor fields of arbitrary rank can be found in Algorithm 1. It selects the most robust rank one tensors and determines the rotation that puts them into the standard position as illustrated in Figure 1.

---

**Algorithm 1:** Moment normalization for tensor fields.

---

- 1: Calculate the moment tensors  $\{T\}$  of the pattern up to order  $o$ , (5).
  - 2: From the tensors  $\{T\}$ , calculate all possible products  $\{\tilde{T}\}$  up to order  $o'$ , (2).
  - 3: From the products  $\{\tilde{T}\}$ , calculate all possible contractions to first  $\{\tilde{T}^1\}$  and second rank  $\{\tilde{T}^2\}$ , (3).
  - 4: Compute all eigenvectors  $\{v\}$  of the symmetric part of the second rank contractions  $\{\tilde{T}^2\}$ .
  - 5: Weigh the eigenvectors  $v_i \in \{v\}$  by the difference of the corresponding eigenvalues  $\tilde{v}_i := v_i(\min_{j \neq i} |\lambda_i - \lambda_j|)$  and add the result to the set  $\{\tilde{T}^1\}$ .
  - 6: Chose  $v_1$  as the contraction or eigenvector with the biggest norm, i.e.  $v_1 := \operatorname{argmax}_{v \in \{\tilde{T}^1\}} \|v\|$ .
  - 7: Chose  $v_2$  as the contraction or eigenvector with the highest component orthogonal to  $v_1$ , i.e.  $v_2 := \operatorname{argmax}_{v \in \{\tilde{T}^1\}} \|v_1 \times v\|$ .
  - 8:  $R_1$  is the rotation matrix around the axis  $v_1 + (1, 0, 0)^T$  by the angle  $\alpha_1 = \pi$ .
  - 9:  $R_2$  is the rotation matrix around  $v_1$  by the angle  $\alpha_2 = -\operatorname{atan2}((R_1 v_2)^3, (R_1 v_2)^2)$ .
  - 10: Rotate the moment tensors by  $R = R_2 R_1$  using the transformation rule (1).
- 

In the two-dimensional case, the rotation matrix  $R$  is simply the one by the angle  $\alpha_1 = -\operatorname{atan2}(v_1^2, v_1^1)$  and the second step is not necessary. The upper indices indicate the vector component.

Please note that the normalization step is only performed on the moments using Equation (1), and not on the field or the pattern themselves. The algorithm does not require any interpolation or sampling.

For the actual pattern detection, we perform the normalization on the pattern, normalize the moments of the field using the same contractions, and determine the similarity using the reciprocal of the Euclidean distance between the moments of the pattern and the field at each position. This process is described in more detail in [4].

In 2D, arbitrary reflections can be generated from the concatenation of a rotation and a reflection at the x-axis. It can be seen from (1) that it leaves the magnitude of

each moment  $T_{j_1 \dots j_m}^{i_1 \dots i_n}$  unchanged but causes a sign change in a component of the tensor if the corresponding indices have an odd number of appearances of twos, which corresponds to the y-direction. This means that we can multiply each component with  $(-1)^{\sum_{i_k=2} 1 + \sum_{j_l=2} 1}$ . If the user wants to normalize with respect to reflections, we proceed as described above for rotations and then simply apply the sign change. For invariance with respect to all orthogonal transformations, we use the maximal similarity of both.

In 3D, the normalization with respect to reflections works analogously. We can combine a rotation with a reflection at the x-y-plane, which leads to a sign change if the number of threes in the indices, corresponding to the z-direction, is odd. Thus we can multiply each component with  $(-1)^{\sum_{i_k=3} 1 + \sum_{j_l=3} 1}$  and proceed as in the 2D case.

In the remainder of this section, we will briefly discuss the complexity of Algorithm 1. The main computational effort goes into the computation of the moments, because we have to evaluate an integral numerically. Assume, we have a  $d$ -dimensional tensor field of rank  $R$  with  $N$  points and want to find a pattern with  $M$  points using moments up to order  $O$ . Then, we need to compute a total of  $\sum_{o=1}^O d^{R+o}$  moments at each of the  $N$  points, each of which requires the evaluation of  $M$  points for the integration, which leaves us with  $NM \sum_{o=1}^O d^{R+o}$  operations. Please note that this step is inherent in all moment-based algorithms. Therefore, they all have a comparable runtime. It can be performed in a preprocessing step, because the moments of the field do not depend on the pattern. Also, the moments do not depend on each other, which enables a straight forward a parallel computation.

The steps 2 to 9 in Algorithm 1 need to be performed on the moments of the pattern only, which makes them independent from the size of the dataset and the size of the pattern. The number of operations depends on the order  $O$  up to which the moments are computed and  $O'$  up to which, we compute the products, which can be considered small compared to the dataset size. For example, we used  $O, O' \leq 5$  in our result section.

Each contraction requires  $d$  operations for each of the  $d^{r-2}$  remaining entries of the tensor. For a tensor of rank  $r$ , there are  $r(r-1)$  possible contractions. The computation of all contractions to first or second order of a tensor of rank  $R$ , takes therefore less than  $\sum_{r=1}^R d^{r-1} r(r-1)$  operations. This process can be accelerated by removing identical contractions that appear because the order in which indices are contracted does not change the result, for example  $\sum_{(1,2)} \sum_{(3,4)} T = \sum_{(1,4)} \sum_{(2,3)} T$ . We also accelerate this process by only considering non-zero moment tensors for the products and the contractions.

Once, we have determined  $v_1$  and  $v_2$ , we can specifically compute only these two products and contractions that are necessary to produce them for each point in the field. The complexity of it depends on the specific  $v_1$  and  $v_2$ .

For the orientation into standard position, Equation (1) needs to be applied to all moment tensors in the field and once to the pattern, which takes  $(N+1)(\sum_{o=1}^O d^{R+o})^2$  operations. Finally, the comparison of the normalized moments of the pattern to the ones of the field requires another  $N(\sum_{o=1}^O d^{R+o})^2$  operations.

The computation of the pattern detection tasks in our result section took less than a minute on a laptop.

## 5 Results

In this section, we apply our algorithm to some use cases to visually demonstrate its effectiveness. We first briefly show how our algorithm improves the normalization of 3D scalar and vector fields by adding more flexibility to the choice of the normalizer thus allowing us to avoid vanishing moments. Then we present results of our method applied to tensor fields in 2D and 3D. We use the first and second derivatives of an analytic scalar field so that the reader can compare the results. Please note that we do not advocate to use higher rank methods to the derivative in cases, where you could as well apply the lower rank algorithm to the original function. The algorithm is meant for pattern detection tasks in tensor fields, like from diffusion or stress measurements, where lower rank data is not available to describe the phenomenon. We do this here only as an illustrative example.

### 5.1 3D Scalar

The normalization of 3D scalar functions in previous approaches has been performed using the second rank moment tensor  $\Sigma = {}^2M$ . The standard position of this symmetric matrix was given by its eigenbasis. That means the rotation that diagonalizes  $\Sigma$  was used as the normalizer. This method is equivalent to aligning the principal axes of the function with the coordinate axes. It fails if  $\Sigma$  does not have three distinguishable eigenvalues. All vectors suggested by Cygansky et al. [9] for normalization rely on the second rank moment tensor, too. As a result, it will not work for patterns without a quadratic component.

Our algorithm is able to compensate for a vanishing  $\Sigma$  by using the contractions of higher rank tensors. To illustrate this, we first consider an analytic use case. Figure 3 shows a cut view of fields that have been generated from different linear combinations of polynomials from first to third degree in  $x$  and  $y$ . We then generate a scalar field such that, at each position  $(i, j, 0)^T \in \mathbb{R}^3, i, j \in \{1, 2, 3\}$ , we center a polynomial given by the formula  $(x-1)^i + 0.5(y-j)^j$  and modulated with the radial Gaussian  $\exp(-4(x-i)^2 - 4(y-j)^2)$ , superimposing the resulting nine functions.

In order to create a pattern for our search, we choose a small section in the lower right corner, rotated and reflected it randomly. This pattern, drawn from one of the nine possibilities, has a linear and a cubic component but no quadratic. A volume rendering of the rotated pattern can be seen in Figure 2. The missing quadratic part causes currently available methods to fail to find a normalizer. Algorithm 1, on the other hand, chooses the first rank tensor  $v_1 = \{^1M\}$  and the contraction of the first two indices in the third rank tensor  $v_2 = \{\Sigma_{(1,2)}^2M\}$  and assumes the correspond-

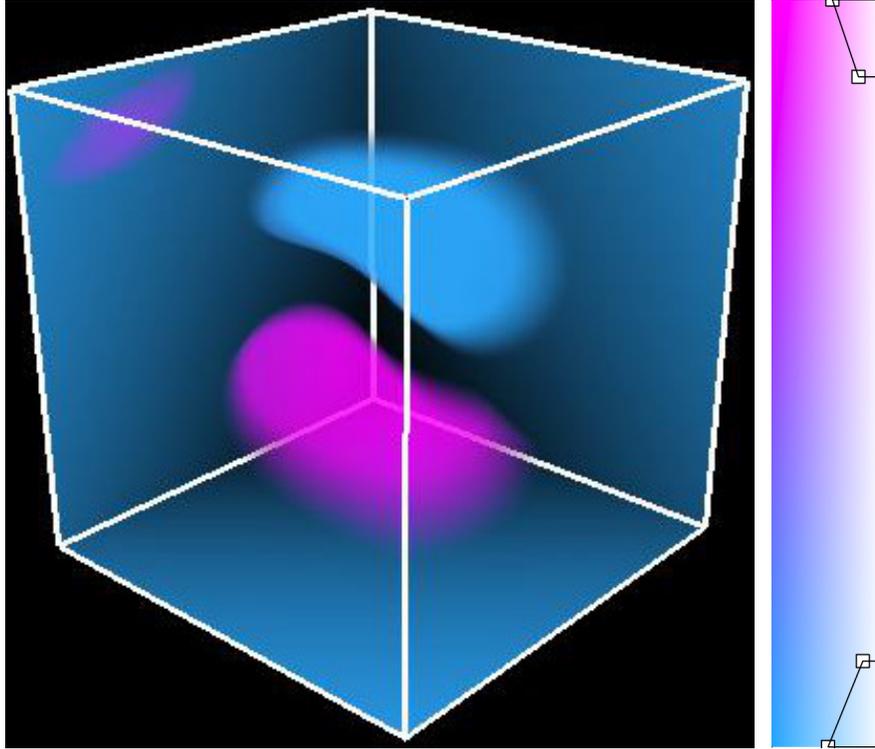


Fig. 2: Volume rendering of the pattern used in Figure 3.

ing standard position. Applying this normalizer, the search results applied to the polynomials of Figure 3 are visualized in the same figure by applying a brightness transfer function (shown to the right of the figure). The brighter the polynomial visualization, the closer it is to the actual search pattern. Algorithm 1 successfully detects the occurrence of the target polynomial in the field. Other less similar structures in the field are indicated with decreasing similarity values. We calculated the moments up to third order, i.e. third rank, and normalized with respect to rotation, reflection, and scaling. Please note that the analytic field was chosen such that it does not depend on  $z$  in order to simplify the visualization. Also, the algorithm was performed on the complete 3D data and the cutting plane is shown for visualization purposes only. The missing  $z$  component is not a simpler special case, but actually more challenging as the algorithm has fewer possible vectors from which to choose.

Please note that the suggested algorithm will coincide with its predecessors if the chosen pattern has a big enough quadratic part. In this case, it is therefore as robust as the former approach. If the pattern lacks this part, the former algorithm will produce an unreliable output, because it will try to determine the orientation of the eigenvectors of a matrix that is numerically zero.

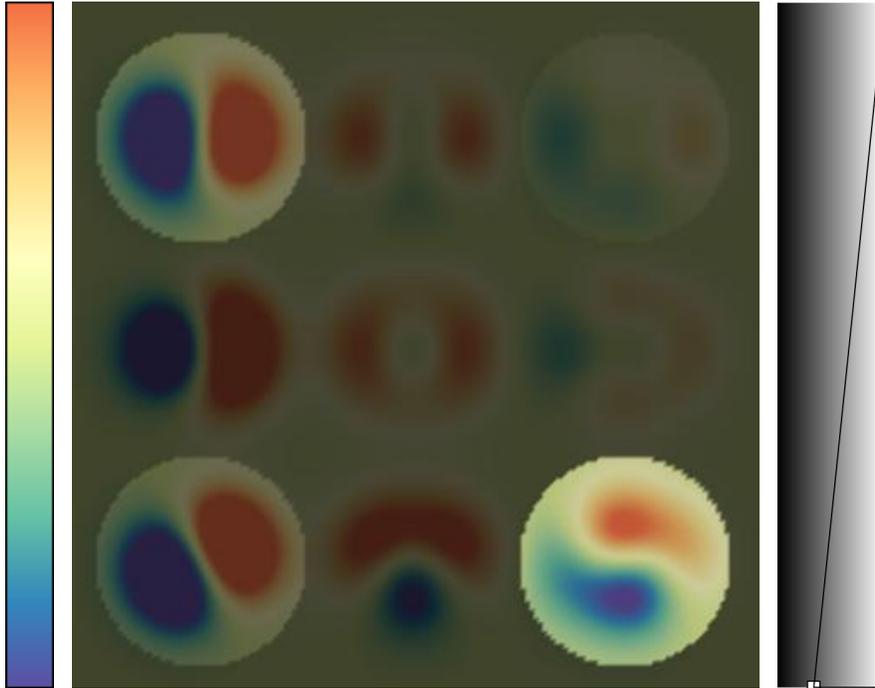


Fig. 3: Color coding of the  $z = 0$  plane of the analytic scalar field. The similarity to the pattern from Figure 2 is encoded using the transfer function on the right. The higher the similarity, the brighter the underlying field.

## 5.2 3D Vector

We now move to the 3D vector case. So far, the normalization of vector field moments makes use of the first order moment tensor  $\Sigma = {}^1M$  which again has second rank as in the scalar case. But in contrast to the latter, this matrix is not symmetric for vector fields. Bujack et al. [5] use the Schur form as standard position. The normalizer is the rotation that transforms  $\Sigma$  into an upper triangular matrix. Analogous to the scalar case, this method fails if  $\Sigma$  is zero. The algorithm in this paper is capable of overcoming this issue because of its flexibility. It can use any first rank contraction and is not bound to a tensor of a specific rank.

We use the gradient of the scalar field from the preceding section as an example 3D vector field and calculate the moments up to third order, i.e. fourth rank. The randomly rotated and reflected pattern has a constant and a quadratic part. Its linear component now vanishes. Therefore, the second rank moment tensor  $\Sigma$  is zero and the method described in [5] does not yield any result. Algorithm 1 on the other hand retrieves one vector from the first rank tensor of the zeroth order moments  $v_1 = {}^0M$  and one from contracting the first and the last index of the third rank tensor of the

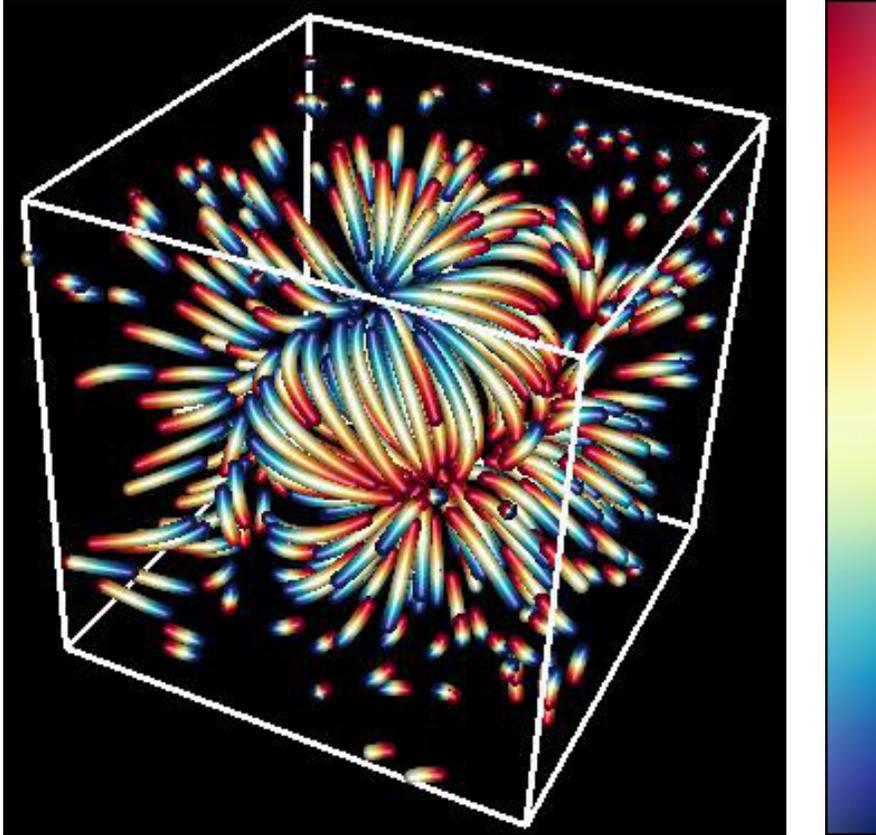


Fig. 4: Stream tubes of the pattern used in Figure 5. Red implies moving forward and blue backward in time.

second order moments  $v_2 = \sum_{(1,3)}^2 M$ . Figure 4 shows streamtubes of the randomly rotated and reflected target pattern and Figure 5 is a visualization of the similarity within the field. The specific similarity values differ from the ones of the scalar field, but the relative order remains the same.

### 5.3 2D Tensor

For the tensor case, we constructed an analytic pattern shown in Figure 6 and a corresponding analytic matrix field in Figure 7, in which we placed the exact copy of the pattern and a squeezed one. Neither occurrence was aligned with the pattern, but the algorithm correctly detected its copy with the highest similarity and the distorted version with lower similarity.

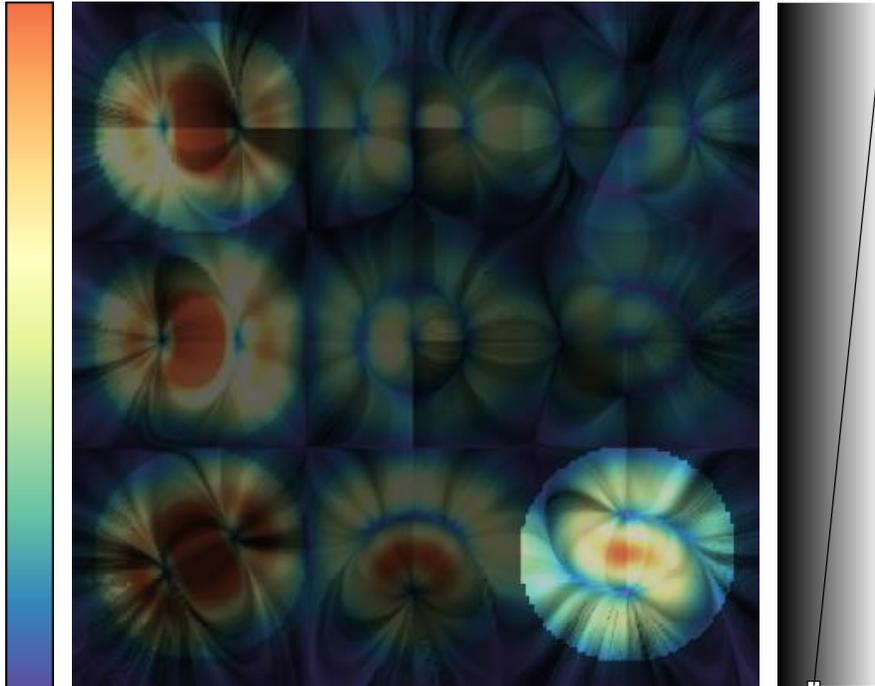


Fig. 5: Cut through the gradient of the analytic scalar field from Figure 3 at the  $z = 0$  plane is visualized with LIC and color encoding the velocity of the vector field. The similarity is encoded using the transfer function on the right. The higher the similarity, the brighter is the underlying field.

The visualization of the pattern matching Figure 7 is done analogously to [4]. We draw circles around the local similarity maxima. The color of the circles encodes the similarity and the diameter represents the integration area for the moment calculation that resulted in the maximum. For comparison, we lay the result of our algorithm on top of a tensor LIC [35] image of the field. This technique is based on the classical line integral convolution (LIC) for vector fields [7]. For symmetric matrix fields, the two LIC images that correspond to the directions of the eigenvectors are calculated and then either one of them or a combined image that interweaves both can be shown. More information about tensor visualization can be found in [24].

To demonstrate the applicability of the algorithm in real world applications, we applied our algorithm to the strain tensor field of the fluid dynamics simulation of the von Kármán vortex street from [2]. The strain tensor is the symmetric part of the Jacobian. It describes the separation of neighboring particles [20].

Figures 8 to 11 show two pattern detection results visualized with tensor LIC [35]. Differently sized patterns were cut out from the field, randomly rotated, and searched for. We use moment tensors up to third order, i.e. fourth rank. The highest

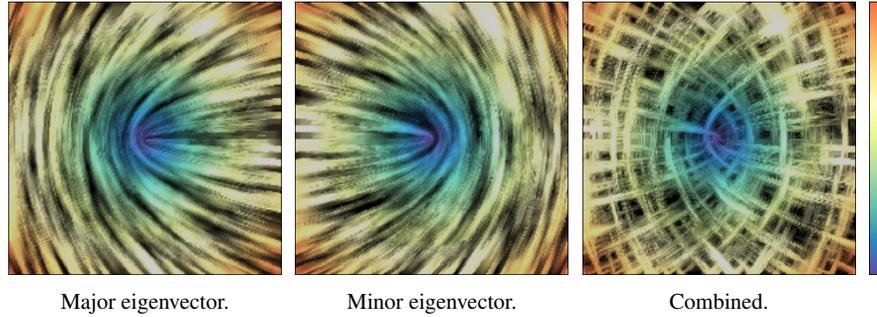


Fig. 6: Tensor LIC of the pattern used in Figure 7. The color map encodes the Frobenius norm of the matrix.

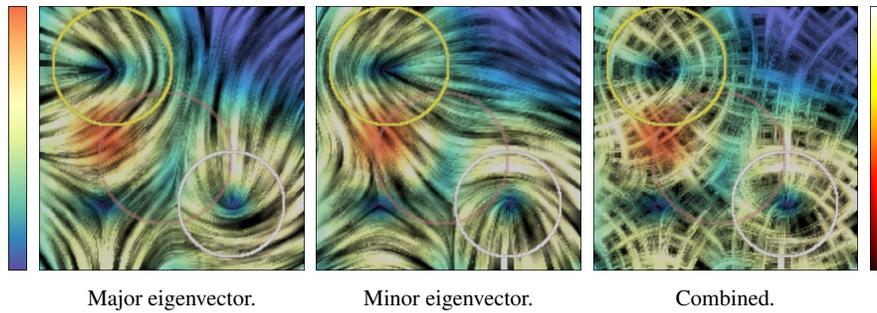


Fig. 7: Output of Algorithm 1 for the pattern from Figure 6 in a matrix field. The color in the LIC corresponds to the Frobenius norm of the matrix, the color of the circles to the similarity.

similarity peak corresponds to the location where the pattern was selected from. The repeating matches nicely show the repetitive structure of the vortex street. In Figure 9, we normalized with respect to rotation, reflection, and scaling. Without the reflection enabled, there are no matches on the lower half of the vortex street. In Figure 11, we normalized with respect to rotation and scaling. If reflection was enabled, additional matches would appear between each two circles.

#### 5.4 3D Tensor

Similar to the 3D vector case, we apply Algorithm 1 to the Hessian, i.e. matrix of the second derivatives, of the scalar field from Figure 3 to consider the 3D tensor case. We use superquadric tensor glyphs [23] and illuminated tensor lines [34, 19, 36] in directions of all three eigenvectors to visualize the randomly rotated and reflected pattern in Figure 12. Figure 13 shows the output of the algorithm. We

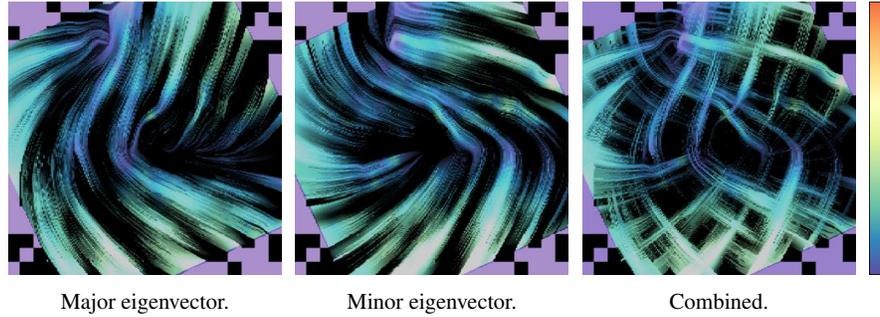


Fig. 8: Tensor LIC of the pattern used in Figure 9. The color map encodes the Frobenius norm of the strain tensor.

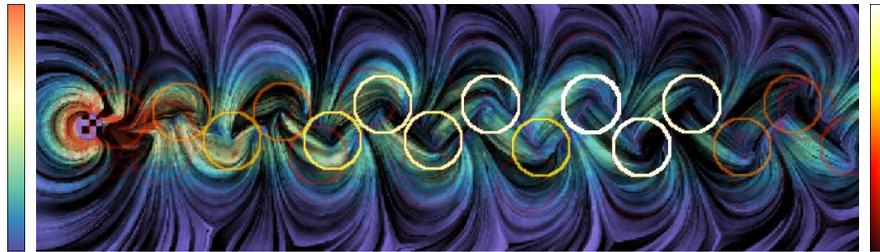


Fig. 9: Output of Algorithm 1 for the pattern from Figure 8 in the strain tensor field of the von Kármán vortex street laid over the tensor LIC of the major eigenvector. The color in the LIC corresponds to the Frobenius norm of the strain tensor, the color of the circles to the similarity.

used moments up to second order, i.e. fourth rank, and normalized with respect to rotation, reflection, and scaling. Again, the position moment that was used to create the target pattern is clearly identified as the strongest match and the locations with lower resemblance follow. The visualization of the similarity is done using tensor LIC [35] color coded with the Frobenius norm of the Laplacian.

## 6 Discussion

In this paper, we have elucidated the properties of the moment tensors of tensor fields of arbitrary dimension and rank in Theorem 2. We have applied this theoretical result to develop an algorithm for the generation of rotation invariants for two- and three-dimensional tensor fields using moment normalization and demonstrated its applicability to analytical data and simulation use cases. To our knowledge this is the first time that moment invariants have been practically applied to detect pat-

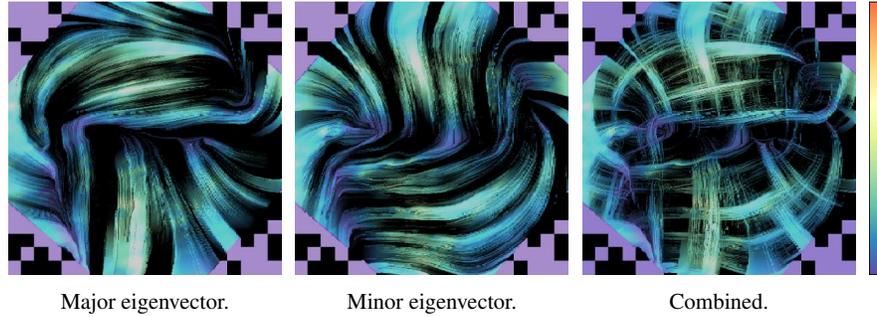


Fig. 10: Tensor LIC of the pattern used in Figure 11. The color map encodes the Frobenius norm of the strain tensor.

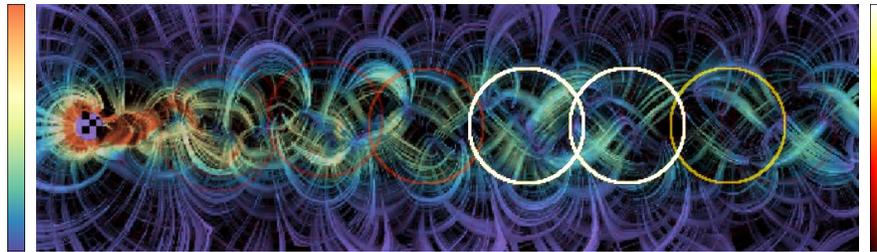


Fig. 11: Output of Algorithm 1 for the pattern from Figure 10 in the strain tensor field of the von Kármán vortex street laid over the tensor LIC of both eigenvectors. The color in the LIC corresponds to the Frobenius norm of the strain tensor, the color of the circles to the similarity.

terns in matrix fields. Further, we have shown how the algorithm improves existing algorithms for 3D scalar fields and 3D vector fields.

We would like to emphasize that the work on this topic is far from finished. The existing algorithms, both the normalization and the generator approach, still have some shortcomings. For the generator approach, the question of redundancy has not been completely solved yet and both approaches struggle to find symmetric patterns. For patterns that are not completely rotationally symmetric but show a certain rotational symmetry, all contractions to zeroth, first, and second rank may be zero and hence both the generator and the normalization approach fail. In 2D this issue has been treated for scalar fields by Flusser and Suk [16], who generated bases specifically for this problem. For vector fields, Bujack et al. [3] solve the problem using a set of multiple standard positions if a rotationally symmetric moment is chosen for the normalization. It needs to be investigated in future work how or if either of these solutions can be generalized to contraction based methods.

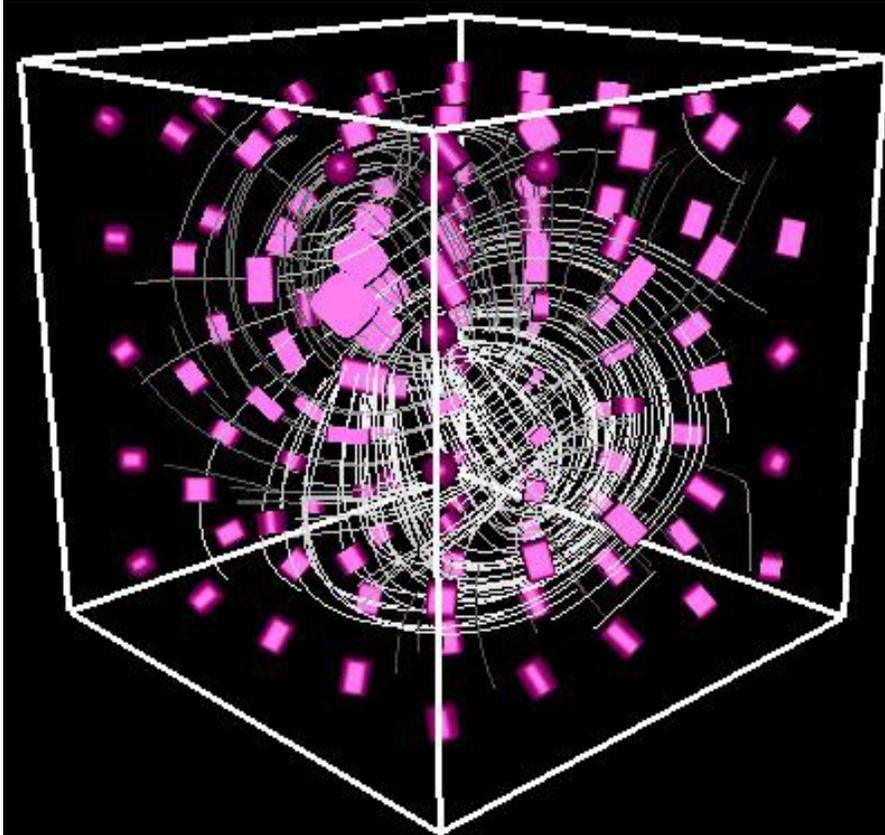


Fig. 12: Superquadric tensor glyphs and illuminated tensor lines of the pattern.

## Acknowledgments

We would like to thank the FAnToM development group from Leipzig University for providing the environment for the visualization of the presented work. Further, we thank Professor Mario Hlawitschka for providing the dataset used in this publication and Terece Turton for help with the writing. This work was partly funded by the German Research Foundation (DFG) as part of the the IRTG 2057 "Physical Modeling for Virtual Manufacturing Systems and Processes".

## References

1. R. Bowen and C. Wang. *Introduction to Vectors and Tensors*. Dover books on mathematics. Dover Publications, 2008.

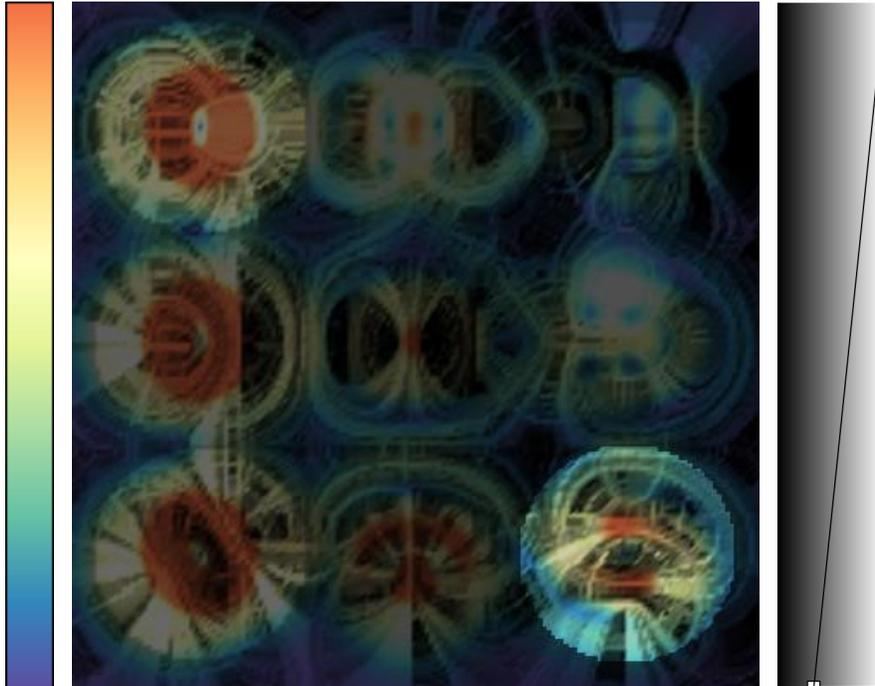


Fig. 13: Cut through the Laplacian of the analytic field from Figure 3 at the  $z = 0$  plane is visualized with tensor LIC. The color bar left encodes the Frobenius norm. The similarity is encoded using the transfer function on the right. The higher the similarity, the brighter is the underlying field.

2. R. Bujack, M. Hlawitschka, and K. I. Joy. Topology-Inspired Galilean Invariant Vector Field Analysis. In *Proceedings of the IEEE Pacific Visualization Symposium, PacificVis 2016 in Taipei, Taiwan*, pages 72–79, 2016.
3. R. Bujack, I. Hotz, G. Scheuermann, and E. Hitzer. Moment Invariants for 2D Flow Fields Using Normalization. In *IEEE Pacific Visualization Symposium, PacificVis 2014 in Yokohama, Japan*, pages 41–48, 2014.
4. R. Bujack, I. Hotz, G. Scheuermann, and E. Hitzer. Moment Invariants for 2D Flow Fields via Normalization in Detail. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 21(8):916–929, Aug 2015.
5. R. Bujack, J. Kasten, I. Hotz, G. Scheuermann, and E. Hitzer. Moment Invariants for 3D Flow Fields via Normalization. In *IEEE Pacific Visualization Symposium, PacificVis 2015 in Hangzhou, China*, 2015.
6. G. Burel and H. Henocq. 3D Invariants and their Application to Object Recognition. *Signal procesing*, 45(1):1–22, 1995.
7. 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.
8. N. Canterakis. Complete moment invariants and pose determination for orthogonal transformations of 3D objects. In *Mustererkennung 1996, 18. DAGM Symposium, Informatik aktuell*, pages 339–350. Springer, 1996.

9. D. Cyganski, S. J. Kreda, and J. A. Orr. Solving for the general linear transformation relating 3-d objects from the minimum moments. In *1988 Robotics Conferences*, pages 204–211. International Society for Optics and Photonics, 1989.
10. D. Cyganski and J. A. Orr. Applications of Tensor Theory to Object Recognition and Orientation Determination. *IEEE Trans. Pattern Anal. Mach. Intell.*, PAMI-7(6):662–673, 1985.
11. H. Dirilten and T. G. Newman. Pattern matching under affine transformations. *Computers, IEEE Transactions on*, 100(3):314–317, 1977.
12. L. Florack, B. T. H. Romeny, J. Koenderink, and M. Viergever. Cartesian differential invariants in scale-space. *Journal of Mathematical Imaging and Vision*, 3(4):327–348, 1993.
13. L. Florack, B. T. H. Romeny, J. J. Koenderink, and M. A. Viergever. General intensity transformations and differential invariants. *Journal of Mathematical Imaging and Vision*, 4(2):171–187, 1994.
14. J. Flusser. On the independence of rotation moment invariants. *Pattern Recognition*, 33(9):1405–1410, 2000.
15. J. Flusser. On the inverse problem of rotation moment invariants. *Pattern Recognition*, 35:3015–3017, 2002.
16. J. Flusser and T. Suk. Rotation Moment Invariants for Recognition of Symmetric Objects. *Image Processing, IEEE Trans.on*, 15(12):3784–3790, 2006.
17. J. Flusser, B. Zitova, and T. Suk. *Moments and moment invariants in pattern recognition*. John Wiley & Sons, 2009.
18. P. Grinfeld. *Introduction to Tensor Analysis and the Calculus of Moving Surfaces*. Springer New York, 2013.
19. M. Hlawitschka and G. Scheuermann. Hot-lines: Tracking lines in higher order tensor fields. In *Proc. IEEE Visualization*, pages 27–34. IEEE, 2005.
20. I. Hotz, L. Feng, H. Hagen, B. Hamann, K. Joy, and B. Jeremic. Physically based methods for tensor field visualization. In *Proc. IEEE Visualization*, pages 123–130. IEEE, 2004.
21. M.-K. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187, 1962.
22. M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation Invariant Spherical Harmonic Representation of 3D Shape Descriptors. In *Symposium on Geometry Processing*, 2003.
23. G. Kindlmann. Superquadric tensor glyphs. In *Proceedings of the Sixth Joint Eurographics-IEEE TCVG conference on Visualization*, pages 147–154. Eurographics Association, 2004.
24. D. H. Laidlaw and J. Weickert. *Visualization and Processing of Tensor Fields: Advances and Perspectives*. Springer Science & Business Media, 2009.
25. M. Langbein and H. Hagen. A generalization of moment invariants on 2d vector fields to tensor fields of arbitrary order and dimension. In *International Symposium on Visual Computing*, pages 1151–1160. Springer, 2009.
26. C. Lo and H. Don. 3-D Moment Forms: Their Construction and Application to Object Identification and Positioning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(10):1053–1064, 1989.
27. Z. Pinjo, D. Cyganski, and J. A. Orr. Determination of 3-D object orientation from projections. *Pattern Recognition Letters*, 3(5):351–356, 1985.
28. T. H. Reiss. The revised fundamental theorem of moment invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):830–834, 1991.
29. M. Schlemmer. *Pattern Recognition for Feature Based and Comparative Visualization*. PhD thesis, Universität Kaiserslautern, Germany, 2011.
30. M. Schlemmer, M. Heringer, F. Morr, I. Hotz, M. Hering-Bertram, C. Garth, W. Kollmann, B. Hamann, and H. Hagen. Moment Invariants for the Analysis of 2D Flow Fields. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1743–1750, 2007.
31. 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.
32. T. Suk and J. Flusser. Tensor Method for Constructing 3D Moment Invariants. In *Computer Analysis of Images and Patterns*, volume 6855 of *Lecture Notes in Computer Science*, pages 212–219. Springer Berlin, Heidelberg, 2011.

33. T. Suk, J. Flusser, and J. Boldyš. 3d rotation invariants by complex moments. *Pattern Recognition*, 48(11):3516–3526, 2015.
34. D. Weinstein, G. Kindlmann, and E. Lundberg. Tensorlines: Advection-diffusion based propagation through diffusion tensor fields. In *Proc. IEEE Visualization*, pages 249–253. IEEE, 1999.
35. X. Zheng and A. Pang. Hyperlic. In *Proc. IEEE Visualization*, pages 249–256. IEEE, 2003.
36. M. Zöckler, D. Stalling, and H.-C. Hege. Interactive visualization of 3d-vector fields using illuminated stream lines. In *Proc. IEEE Visualization*, pages 107–113. IEEE, 1996.