Appeared in *Proceedings of IEEE Visualization '99*, pp. 263-270, 1999.

The "Parallel Vectors" Operator - A Vector Field Visualization Primitive

Ronald Peikert and Martin Roth

Computer Graphics Research Group
Dept. of Computer Science, ETH Zürich, Switzerland
peikert@inf.ethz.ch, roth@acm.org

Abstract

In this paper we propose an elementary operation on a pair of vector fields as a building block for defining and computing global line–type features of vector or scalar fields. While usual feature definitions often are procedural and therefore implicit, our operator allows precise mathematical definitions. It can serve as a basis for comparing feature definitions and for reuse of algorithms and implementations. Applications focus on vortex core methods.

1 Introduction

During recent years, feature extraction techniques have become increasingly interesting as a complementary way to visualize scalar or vector fields.

Feature extraction in scientific visualization is traditionally described in a rather procedural type of language. This is in contrast to basic visualization techniques such as slices, isosurfaces, integral curves (streamlines) or stream functions. These can be specified by simple algebraic equations or ordinary differential equations, so there is no question that one can carefully distinguish between the mathematical definition and the extraction algorithm. The same level of abstraction has not yet been reached in the area of feature extraction.

As a step towards this goal, we propose an *operator* which for two given vector fields returns the loci of points where the two vectors are parallel. The resulting features in general are *lines*, as we will show. In terms of the classification scheme introduced by [9], these features are *global* and have a spatial dimension of *one*. They are also inherent to the data, in the sense that they are definable without parameters.

In the following, we first review related work in section 2. Section 3 provides definitions and shows that some fundamental properties of vector or scalar fields can be expressed with this operator. Based on this, we show in section 4 that various feature line extraction methods can be reformulated using the "parallel vectors" operator. A collection of nine such applications will be discussed. In section 5 we deal with the problem of unwanted spurious solutions, and section 6 briefly describes some possible ways of implementing the operator.

2 Related work

One area which can benefit from feature extraction is *flow visualization*, in particular the analysis of 3D vector fields. Recent research has focussed on *vortices* which are among the most important features occurring in CFD field data. Vortex *cores* [1, 14, 18, 20, 23, 24] and vortex *hulls* [10, 17] can be rendered as lines and surfaces, but beyond that they can be used to improve other types of visualizations. For example, particles can be started near the core, traced forward and backward and then animated, fading them in or out at the vortex boundary. This way, a single vortex is visualized. Another usage is to extract a single feature for interactive access. And finally, features can be tracked over time [22] or while changing design parameters.

In the context of *geomorphology*, Finsterwalder [6] defines *ridge* and *valley lines:* A point is on a ridge or valley line if the slope is minimal along the contour line. In other words, ridges and valleys are where contour lines lie maximally apart. A mathematical formulation of this definition was already given by De Saint-Venant [5] in 1852. Ridge and valley lines are used also in the field of *computer vision* [8]. Here, grayscale images are regarded as height fields.

A different usage of the term *ridge* has been established in *differential geometry*: For a surface in 3-space, *ridges and ravines* are defined as the loci of points of maximal normal curvature along the curvature line in the direction of maximal curvature (see e.g. [2]). In contrast to topographic ridge lines, this definition is *isotropic*, i.e. invariant under global rotations. Isotropy is a desirable property for purposes such as feature line extraction on arbitrary surfaces in 3-space [11, 15]. However, if the surface is a height field, this is not the case: Ridge and valley lines are expected to change when the whole terrain is tilted.

There also exist line-type features of 2D data which become surface-type features for 3D data. In contrast to true 1D features, such $(n-1)$*-dimensional* features cannot be specified with our operator. An example in image processing are *edges*, which can be extracted by finding zero-crossings of the Laplacian (e.g. [8]). The same method applied to 3D data results in surfaces, used e.g. in volume rendering [28].

3 Vector parallelism

3.1 Definitions and notation

As we will frequently switch between scalar and vector fields and between dimensions, we need a short notation for the various types of fields: An n-dimensional (or nD) *scalar field* is a real-valued function of n spatial variables, likewise an nD *vector field* is an n-component vector function of n spatial variables.

Appeared in *Proceedings of IEEE Visualization '99*, pp. 263-270, 1999.

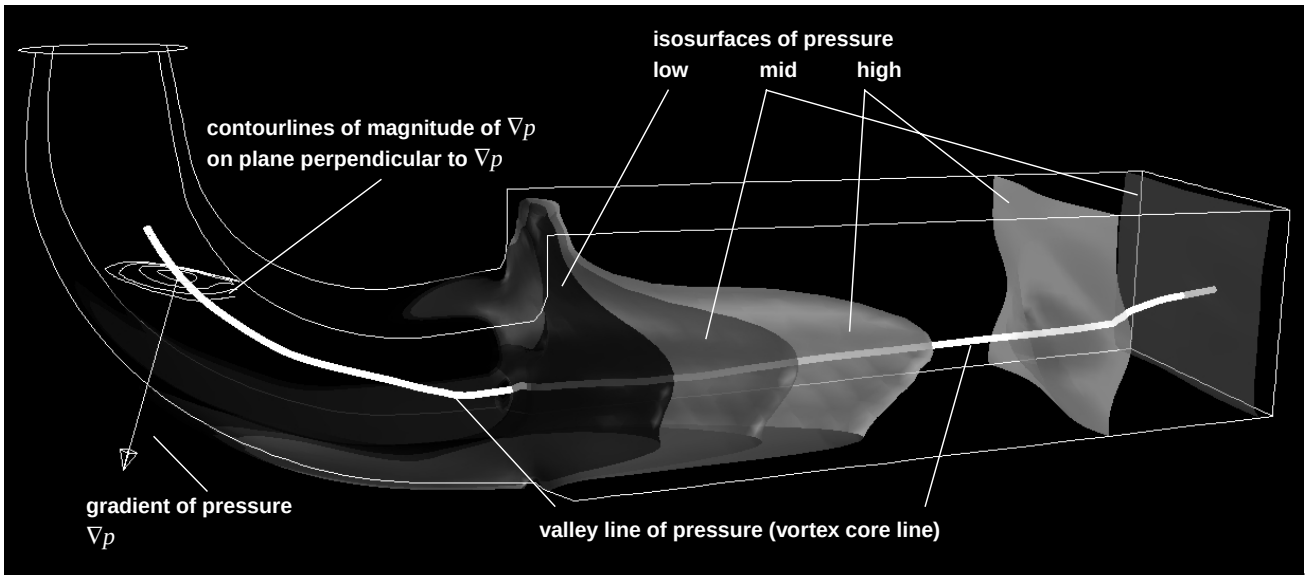

Figure 1: A valley line *(see section 3.3) of pressure extracted from simulation data for a draft tube of a Francis turbine. This feature indicating the center of swirl cannot be visualized using pressure* isosurfaces *because of the global pressure change along the tube.*

Let $\boldsymbol{v}$ and $\boldsymbol{w}$ be n-dimensional vector fields. Then

$$S = \{\boldsymbol{x} : \boldsymbol{v}(\boldsymbol{x}) = 0\} \cup \{\boldsymbol{x} : \exists\lambda, \boldsymbol{w}(\boldsymbol{x}) = \lambda\boldsymbol{v}(\boldsymbol{x})\} \qquad (1)$$

is the set of locations where either the two vectors have the same direction or one of them is zero. The first term of eq. (1) is necessary to avoid $\lambda = \pm\infty$. The second term consists of n scalar equations for the $n+1$ unknowns $x_1, \ldots, x_n, \lambda$. In the nondegenerate case (of independent equations), solutions are one-dimensional. This can be seen by the following argument.

At each point $\boldsymbol{x}$, the vector $\boldsymbol{w}(\boldsymbol{x})$ can be decomposed into a component $\boldsymbol{w}_{\parallel}$ parallel to $\boldsymbol{v}(\boldsymbol{x})$ and a component $\boldsymbol{w}_{\perp}$ perpendicular to $\boldsymbol{v}(\boldsymbol{x})$. The condition that $\boldsymbol{v}(\boldsymbol{x})$ and $\boldsymbol{w}(\boldsymbol{x})$ are parallel means that $\boldsymbol{w}_{\perp}$ is zero, which restricts only $n-1$ of the n degrees of freedom.

If n is either 2 or 3, the set can alternatively be expressed as

$$S = \{\boldsymbol{x} : \boldsymbol{v}(\boldsymbol{x}) \times \boldsymbol{w}(\boldsymbol{x}) = 0\} \qquad (2)$$

(where the cross product of two 2-vectors is just the scalar $v_1 w_2 - v_2 w_1$).

If $n = 3$, this definition involves three scalar equations for three unknowns. However, these equations are not independent. This is the reason why it is misleading to consider $\boldsymbol{v} \times \boldsymbol{w}$ as a *derived field* for the purpose of finding the set of its zeros: Such a field is degenerate in the sense that the zeros in general are not isolated points but lines, which prohibits application of standard methods.

Definition: For two given n-dimensional vector fields $\boldsymbol{v}$ and $\boldsymbol{w}$, we denote by $\boldsymbol{v} \parallel \boldsymbol{w}$ ("$\boldsymbol{v}$ parallel $\boldsymbol{w}$") the *operator* which returns the set S (eq. (1)) restricted to nondegenerate solutions. In other words, we exclude isolated points and points which have a 2D neighborhood in S.

It can easily be seen that the lines of $\boldsymbol{v} \parallel \boldsymbol{w}$ are *closed curves* and in general do not intersect: In 2D, they are 0-isolines of the cross product and therefore have these properties. In 3D, the cross product has three scalar components, hence there are three 0-isosurfaces. As shown above, their intersection is a set of lines rather than isolated points. And finally, because each isosurface is closed, so is their intersection.

The two vector fields $\boldsymbol{v}$ and $\boldsymbol{w}$ can be given independently, however, in our applications, $\boldsymbol{w}$ is often derived from $\boldsymbol{v}$. In the remainder of this section, we will examine two such cases.

3.2 The loci of zero curvature

For a C^1 vector field $\boldsymbol{v}$ we can compute its gradient $\nabla\boldsymbol{v}$ (the Jacobian, which is a *matrix* field) and then examine the set

$$\boldsymbol{v} \parallel (\nabla\boldsymbol{v})\,\boldsymbol{v} \qquad (3)$$

If we interpret $\boldsymbol{v}$ as a steady velocity field, then for each location $\boldsymbol{x}$, the right hand side is exactly the *acceleration* $D\boldsymbol{v}/Dt$ of a particle at $\boldsymbol{x}$. If now the acceleration is parallel to the velocity, this implies that the local streamline curvature is zero. Therefore, eq. (3) is the condition that the streamline passing through point $\boldsymbol{x}$ has zero curvature at $\boldsymbol{x}$.

We also note that for a solution $\boldsymbol{x}$ of eq. (3), $\boldsymbol{v}(\boldsymbol{x})$ is an *eigenvector* of $\nabla\boldsymbol{v}(\boldsymbol{x})$. In the following, we will denote the normalized i-th eigenvector by ε_i and the corresponding eigenvalue by λ_i and choose the numbering such that ε_0 is $\boldsymbol{v}(\boldsymbol{x})$.

3.3 Topographic ridge and valley lines

Given a scalar field p, it is an elementary fact that a necessary condition for an inner local minimum or maximum is $\nabla p = 0$. Less obvious is how to relax this condition to make the general type of solution a set of lines instead of a set of points. This problem is best illustrated by looking at a 2D height field as e.g. a topographical terrain. Let $\boldsymbol{v}$ be the vector field ∇p. The *slope* of the terrain p is then the vector magnitude $\|\boldsymbol{v}\|$ and the *slope lines* (or lines of steepest descent) of p are the *trajectories* of $\boldsymbol{v}$.

Definition: A *topographic ridge or valley line* is the set of points where the slope is locally minimal compared to points of the same elevation.

The distinction between ridge lines and valley lines can be made by looking at a profile taken in the direction normal to $\boldsymbol{v}$. A more formal definition can be based on the eigenvalues of the elevation Hessian $\boldsymbol{H}$ as will be shown in section 3.5. For points on ridge or valley lines, one eigenvector of $\boldsymbol{H}$ is aligned with the elevation gradient and the other is tangential to the contour line. The eigenvalue associated with the latter discriminates ridges (negative sign) from valleys (positive sign).

An equivalent formulation of this definition is commonly used in computer vision. As Haralick [7] describes it, points on a ridge or valley line can be found by walking in the direction of greatest magnitude of the second directional derivative. The ridge peak or valley bottom occurs when the first directional derivative has a zero crossing.

Some confusion about the correct definition of topographic ridge and valley lines has appeared in the literature. A number of methods produce only *approximate* solutions. In [12], *elevation maxima* in *axis-parallel sections* are chosen. Other authors (e.g. [27]) search for *maximum curvature of contour lines*. Lines generated by both these methods can differ visibly from correct extremum lines. A simple example for this is the *inclined elliptical cylinder* already given in the 19th century by [29]. Fig. 2 shows this example in both top view and perspective view. The ellipse is aligned with the image axes, therefore the marked points are the elevation maxima in sections of constant x_1 and at the same time the contour curvature maxima.

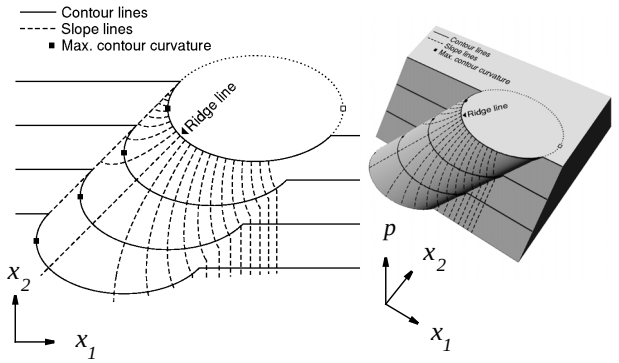

Figure 2: Left: Top view of the inclined elliptical cylinder example. Right: Perspective view of the same height field. The ridge line does not pass through the points of maximal contour curvature.

Another approximation is to extract the *critical lines* [16], i.e. the slope lines (lines of steepest descent) connecting passes (saddles) with peaks or pits. In general, however, a ridge or valley line is not a slope line: Breton de Champs [4] proved in 1854 that it is only if its ground projection is a straight line. But since in typical terrain data, ridge and valley lines are often close enough to slope lines, this is a reasonable approximation.

3.4 Extremum lines

For a vector field $\boldsymbol{v}$ and a given point $\boldsymbol{x}$ let $N_{\boldsymbol{x},\boldsymbol{v}}$ denote the *normal space* of $\boldsymbol{v}$ at $\boldsymbol{x}$, i.e. the line or (hyper-)plane containing $\boldsymbol{x}$ and being normal to $\boldsymbol{v}$.

Definition: An *extremum line* [25] of a vector field $\boldsymbol{v}$ can then be defined as the set of points $\boldsymbol{x}$ where the restriction of $\|\boldsymbol{v}\|$ to $N_{\boldsymbol{x},\boldsymbol{v}}$ is locally extremal. It is called a *minimum (maximum) line* if $\|\boldsymbol{v}\|$ is minimal (maximal).

In 3D, this means that the vector magnitude is minimal or maximal at $\boldsymbol{x}$ w.r.t. the plane through the point $\boldsymbol{x}$ and perpendicular to $\boldsymbol{v}(\boldsymbol{x})$.

Of course, $\|\boldsymbol{v}\|$ can be replaced by $\boldsymbol{v}^2$ which leads to the necessary condition

$$\boldsymbol{v} \parallel \nabla(\boldsymbol{v}^2) \tag{4}$$

for extremum lines. In 3D (or higher dimensions), solutions of eq. (4) include "saddle lines", consisting of points where $\|\boldsymbol{v}\|$ assumes a saddle in $N_{\boldsymbol{x},\boldsymbol{v}}$.

By applying the identity $\nabla(\boldsymbol{v}^2) = 2(\nabla\boldsymbol{v})^{\mathsf{T}}\boldsymbol{v}$ to eq. (4) we obtain the equivalent necessary condition for extremum lines,

$$\boldsymbol{v} \parallel (\nabla\boldsymbol{v})^{\mathsf{T}}\boldsymbol{v} \tag{5}$$

which differs from the zero curvature condition eq. (3) only by the matrix transposition.

3.5 Extremum lines of conservative vector fields

If a vector field $\boldsymbol{v}$ is *conservative*, it can be written as ∇p for some scalar field p which is called a *potential* for $\boldsymbol{v}$. For conservative vector fields, the *minimum* lines are of particular interest because they contain ridge and valley lines as special cases. This follows immediately from the definitions:

Observation 1: Topographic ridge (valley) lines are minimum lines of $\boldsymbol{v} = \nabla p$ with the additional property that each point $\boldsymbol{x}$ is a local maximum (minimum) of p restricted to $N_{\boldsymbol{x},\boldsymbol{v}}$.

Observation 1 can be used as a definition of ridge and valley lines in n dimensions. Note that by definition of $N_{\boldsymbol{x},\boldsymbol{v}}$, the restricted function p automatically assumes an extremum or a saddle in $\boldsymbol{x}$.

For a conservative $\boldsymbol{v}$, the matrix $\nabla\boldsymbol{v}$ is *symmetric* and therefore eq. (3) and eq. (5) are identical:

Observation 2: In a conservative vector field, the set of zero curvature lines is equal to the union of extremum lines and saddle lines.

Principal axes formulation

For the next observation, we look at the second-order Taylor approximation of p around a point $\boldsymbol{x}_0$ of an extremum or saddle line:

$$p(\boldsymbol{x}_0 + \boldsymbol{x}) = p(\boldsymbol{x}_0) + \boldsymbol{v}(\boldsymbol{x}_0)\cdot\boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^{\mathsf{T}}\boldsymbol{H}\boldsymbol{x} + O(\|\boldsymbol{x}\|^3) \tag{6}$$

Here, $\boldsymbol{v}$ denotes ∇p and $\boldsymbol{H}$ denotes $\nabla\boldsymbol{v}$ (which is the Hessian of p) at $\boldsymbol{x}_0$. We already know that on an extremum or saddle

line, $\boldsymbol{v}$ must be an eigenvector of $\boldsymbol{H}$. The symmetry of $\boldsymbol{H}$ causes the eigenvalues to be real and the eigenvectors to be orthogonal.

What we want to do next is formulate conditions for topographic ridge lines in terms of the eigenvalues. The conditions can be formally derived from the definition of a ridge, but we believe that the following informal analysis gives more insight.

In the 2D case, Dupin's indicatrix (the intersection with an infinitesimally offset tangential plane) of the surface $z = p(x, y)$ can be constructed. This is either an ellipse or a hyperbola and, when viewed as a curve lying in the x-y-plane, it only depends on the quadratic term, i.e. on $\boldsymbol{H}$. Its center is $\boldsymbol{x}_0$ and its principal axes are proportional to $|\lambda_i|^{-1}\varepsilon_i$. In 3D, these ellipses and hyperbolas become ellipsoids and hyperboloids.

The sign of an eigenvalue determines, for the corresponding principal axis, whether p is below or above its linear approximation when moving away from $\boldsymbol{x}_0$. With this, the conditions for a ridge or valley line are intuitive:

In the case of an ellipse (ellipsoid), we are on a ridge line if all eigenvalues are negative and if λ_0 has smallest absolute value, and therefore ε_0 is the direction of the largest principal axis. The indexing is again such that ε_0 is the eigenvector aligned with $\boldsymbol{v}(\boldsymbol{x}_0)$.

In the case of a hyperbola (hyperboloid), we are on a ridge if λ_0 is positive and the other eigenvalue(s) are negative.

The conditions for valley lines are obtained by changing signs. Combining the cases, we get for a ridge line:

$$\lambda_i < 0, \qquad \lambda_i < \lambda_0 \tag{7}$$

and for a valley line:

$$\lambda_i > 0, \qquad \lambda_i > \lambda_0 \tag{8}$$

where $i = 1$ in 2D and $i = 1, 2$ in 3D. Therefore, given a set of extremum or saddle lines (eq. (5)), we can select ridge lines by imposing eq. (7) and valley lines by eq. (8). The resulting lines are minimum lines according to observation 1.

4 Applications

4.1 Zero curvature loci

❶ Vortex cores defined by the velocity gradient

Sujudi and Haimes [24] defined vortex cores in 3D flows based on eigenvectors of the Jacobian. In [19] we showed that their definition is equivalent to eq. (3) combined with the condition that the matrix $\nabla\boldsymbol{v}$ has a complex pair of eigenvalues. The latter can be tested by checking the characteristic polynomial for a negative discriminant.

❷ Separation and attachment lines

Similarly, Kenwright's separation line criterion [13] can be formulated as eq. (3), combined with the condition $\lambda_0\lambda_1 < 0$ or $|\lambda_0| < |\lambda_1|$ for the eigenvalues of $\nabla\boldsymbol{v}$. This time, $\boldsymbol{v}$ is a 2D vector field, namely the velocity projected onto a solid surface. That both eigenvalues of $\nabla\boldsymbol{v}$ are real numbers follows from eq. (3), which asserts a real eigenvector.

For Fig. 3, a slice of the "bluntfin" data next to the solid surface has been extracted. The projected vector field $\boldsymbol{v}$ is shown by white arrows in Fig. 3a. The black arrows indicate the field $\boldsymbol{w} = (\nabla\boldsymbol{v})\,\boldsymbol{v}$. The set $\boldsymbol{v} \parallel \boldsymbol{w}$ is shown as the thick lines where the black lines are those which meet the angle criterion explained in section 5. Fig. 3b demonstrates the zero curvature property of the separation and attachment lines: it can be observed that the thick lines consist of inflection points of the streamlines of $\boldsymbol{v}$ (the skin friction lines).

In the rest of this section, we will find more such combinations of a parallelism condition and a Boolean selection criterion. The selection criteria used in the above two applications are quite typical. In the first case, it is the sign of an algebraic expression, and in the second case it is based on the relative size of the eigenvalue associated with the eigenvector $\boldsymbol{v}$. We will first concentrate on the parallelism conditions and postpone the discussion of selection criteria to section 5.

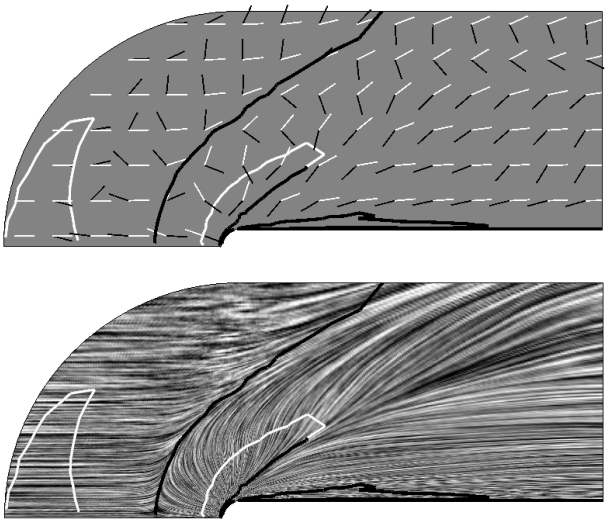

Figure 3: a) Separation and attachment lines on a solid surface of the "bluntfin" data set (black lines). The white arrows indicate the projected velocity $\boldsymbol{v}$, the black arrows the field $\boldsymbol{w} = (\nabla\boldsymbol{v})\,\boldsymbol{v}$. b) The same lines compared to an LIC of $\boldsymbol{v}$.

4.2 Ridge and valley lines

❸ Geomorphology and computer vision

Topographic ridge and valley lines are concepts used in geomorphology [6] and computer vision [7]. Lines are solutions of eq. (5), but practical applications require careful filtering of the input field and post-processing [3] of the output lines.

In image processing applications, ridges are often used for structure enhancement. Another example is the *medial axis transform*, which can be expressed as the ridges of the signed distance map.

❹ Vortex cores defined by minimal pressure

Miura and Kida [18] describe a method for extracting vortex cores based on a pressure-like scalar quantity p which is computed from the velocity field. In our terminology we can say that their procedure constructs the *valley lines* of the field p. An example is shown in Fig. 1.The algorithm of Miura and Kida

steps along these lines while performing local corrections. These ensure that the gradient of p is an eigenvector of the Hessian of p. Other lines than valley lines are excluded by applying eq. (8) after each step.

4.3 Extremum lines

❺ Minimum and maximum lines

Tang and Medioni [25] show that many problems arising in medical imaging or digital terrain models consist of finding extremum lines. Such problems can therefore be expressed by eq. (4) or eq. (5), usually constrained to either the minimum or maximum lines.

❻ Maximum lines of vorticity

Strawn et al. [23] defined a vortex core as the set of points where *vorticity magnitude* is locally maximal on the plane $N_{\boldsymbol{x},\,\omega}$ normal to the vorticity vector. Again, this condition can be expressed by eq. (4), this time applied to the vorticity field.

4.4 General vector parallelism

❼ Vortex cores defined by normalized helicity

Helicity of a vector field $\boldsymbol{v}$ is defined as $\boldsymbol{v} \cdot (\nabla \times \boldsymbol{v})$, while *normalized helicity* is $(\boldsymbol{v}/\|\boldsymbol{v}\|) \cdot (\nabla \times \boldsymbol{v}/\|\nabla \times \boldsymbol{v}\|)$, which is just the cosine of the angle between $\boldsymbol{v}$ and its curl $\nabla \times \boldsymbol{v}$. A vortex core according to Levy et al. [14] is indicated by a large absolute value of normalized helicity.

We can again make use of the "parallel vectors" operator by observing that normalized helicity of ± 1 is equivalent to $\boldsymbol{v} \parallel \nabla \times \boldsymbol{v}$.

A previous method for extracting this type of vortex cores is based on isosurfaces of normalized helicity. But, due to the discretization, levels cannot be chosen as ± 1. Therefore, no clear line structures can be produced.

❽ Vortex cores defined by vorticity and pressure

In [1], Banks and Singer presented a method to first find vortex core lines and then grow a vortex region around them. The core-finding part basically repeats the following sequence:

- Take a step along the vorticity vector.
- At the new point, evaluate the vorticity vector and construct its normal plane.
- In this plane, correct the new point by setting it to the nearest local pressure minimum. Stop if the correction vector exceeds a size limit.

A starting point is found by performing a number of initial steps for establishing convergence.

This procedure is obviously based on the assumption that the direction of the vorticity vector has little variation compared to the variation of pressure. Under this assumption and for small step sizes, the algorithm yields points where vorticity is approximately parallel to the pressure gradient, i.e. $\nabla \times \boldsymbol{v} \parallel \nabla p$.

As usual, the parallelism formulation has additional spurious solutions. In one kind, the pressure is maximal instead of minimal. There is also the case of lines which satisfy $\nabla \times \boldsymbol{v} \parallel \nabla p$ but where the points generated by the algorithm diverge from the line.

❾ Higher-order definition of vortex cores

In [20] we proposed a vortex core definition which better accounts for slowly rotating curved vortices. The parallelism part of the definition is $\boldsymbol{v} \parallel (\nabla \boldsymbol{a})\boldsymbol{v}$ where $\boldsymbol{a}$ is the acceleration $(\nabla \boldsymbol{v})\,\boldsymbol{v}$.

	Author(s)	Problem	Dim.	Type *)	Formula	Criteria
❶	Sujudi/Haimes	vortex cores	3	Z	eq. (3) for velocity	negative discriminant of $\nabla \boldsymbol{v}$ (complex eigenvalue pair)
❷	Kenwright	separation lines	2	Z	eq. (3) for projected velocity	for eigenvalues of $\nabla \boldsymbol{v}$: $\lambda_0\lambda_1 < 0$ or $\lvert\lambda_0\rvert < \lvert\lambda_1\rvert$
❸	Haralick	ridge, valley lines	2	Z,E	eq. (3) or (5) for elevation	ridge line: eq. (7), valley line: eq. (8)
❹	Miura/Kida	vortex cores	3	Z,E	eq. (3) or (5) for pressure-like quantity	valley line: eq. (8)
❺	Tang/Medioni	various	2,3	E	eq. (5) for general vector field	minimum and maximum lines
❻	Strawn/Kenwright/ Ahmad	vortex cores	3	E	eq. (5) for vorticity	maximum lines
❼	Levy/Degani/Seginer	vortex cores	3	G	$\boldsymbol{v} \parallel \nabla \times \boldsymbol{v}$	none
❽	Banks/Singer	vortex cores	3	G	$\nabla \times \boldsymbol{v} \parallel \nabla p$	*p minimal* in plane orthogonal to $\nabla \times \boldsymbol{v}$ (and convergence of the algorithm)
❾	Roth/Peikert	vortex cores	3	G	$\boldsymbol{v} \parallel (\nabla \boldsymbol{a})\boldsymbol{v}$	two complex eigenvalues after projection; angle-based quality

Table 1: Applications of vector parallelism.
**) Z: zero curvature, E: extremum lines, G: general parallelism*

5 Criteria for selecting lines

Many vortex core algorithms in their original formulation use *a priori* tests on the grid nodes (e.g. a discriminant). If a node fails the test, all adjacent cells are ignored by the line extraction procedure.

However, we propose to first extract the lines and then selectively remove line segments based on criteria evaluated at the points of the line. Advantages of this scheme are:

- The "parallel vectors" operator can be implemented as a procedure operating on two arbitrary vector fields. It is implemented once and then used for many purposes.
- Criteria can be applied tolerantly, e.g. accepting single exceptions. We found that this produces more consistent lines.

Whether efficiency is gained or lost depends on the time needed to evaluate criteria and to find parallel vectors.

Types of criteria

The criteria used in the nine above applications are summarized in Table 1. They typically use information from the gradient tensor, such as its discriminant or eigenvalues.

In conservative vector fields, criteria for ridge or valley lines are eq. (7) and eq. (8), respectively.

In nonconservative vector fields, minimum and maximum lines can be discriminated by sampling $\|\boldsymbol{v}\|$ a few times on the line (plane) normal to $\boldsymbol{v}$. This way, calculation of second derivatives can be avoided.

Many applications of the "parallel vectors" operator are based on a model assuming that the resulting line is a trajectory of one of the vector fields. Of course, it would then be a trajectory of the other vector field, too.

However, this assumption is not true in general, therefore the *angle* between the generated line and the vector field can be used as a measure for how well the model fits the data. Limiting this angle below a certain threshold is a useful technique for selecting meaningful lines.

6 Algorithms

6.1 The "parallel vectors" operator

In 2D, the "parallel vectors" operator is easily implemented by computing zero level isolines of the cross product. However, if simple bilinear interpolation of cells is used, then no higher-order singularities can be represented. Hence, it is not possible to obtain e.g. a junction of three ridges in the interior of a cell. For higher quality output, an algorithm such as in [7] can be used.

In the following, we assume vector fields in 3D. Unlike in 2D, there is no obvious implementation of the "parallel vectors" operator, but rather a variety of possible approaches. Since, in implicit form, the "parallel vectors" operator has been used several times in the literature, algorithms can be derived from these publications. They can be grouped into the following four schemes. The choice becomes a trade-off between conflicting goals such as symmetry, completeness or smoothness.

Finding parallel vectors based on isosurfaces

The 3D cross product has three components. If it is zero, then any two of the three projections along the coordinate axes are zero, but not vice versa. Therefore, one could generate isosurfaces for two of the components and find their intersection, e.g. by using the "Marching Lines" algorithm [26]. On the intersection curve, one then has to verify that the third component is zero as well.

However, there are a couple of numerical problems associated with this approach. One is that testing for a zero value must be done tolerantly. And besides this, choosing a fixed pair of components can cause small angles when intersecting isosurfaces. These can be avoided by adaptively choosing the best pair of isosurface for each cell. But then care has to be taken to preserve continuity.

Solution based on Newton iterations within grid faces

This method traverses all faces (triangle or quadrilateral) of grid cells looking for zeros of the vector valued function $\boldsymbol{v} \times \boldsymbol{w}$. Starting in the center of each face, a number of 2D Newton-Raphson steps are performed. If a zero is found but lies outside the current face, it is discarded. Otherwise it is an intersection point of the solution set with the face. Points lying on faces of the same cell are then connected by straight lines. If there are more than two intersection points, the correct topology must be identified, e.g. by subdividing the cell or by estimating the curve tangents.

This method can handle vector fields defined on the cell face by an arbitrary interpolation function, for example, a bilinear interpolation on quadrilaterals.

Analytic solution for triangular faces

A similar method calculates the analytic solution, but assumes linear interpolation on the face. Therefore, quadrilateral faces have to be subdivided into triangles. The search for points where two linear vector fields $\boldsymbol{v}$ and $\boldsymbol{w}$ are parallel can be reduced to an eigenvector problem in the following way: On a triangular face, a linear 3-component vector field $\boldsymbol{v}$ can be written as a function of local triangle coordinates s, t:

$$\boldsymbol{v} = \boldsymbol{V} \begin{bmatrix} s \\ t \\ 1 \end{bmatrix} \tag{9}$$

where $\boldsymbol{V}$ is a 3 by 3 matrix, and analogously for $\boldsymbol{w}$ and $\boldsymbol{W}$. The three coordinates can be interpreted as homogeneous coordinates s,t,u with u set to one.

Two fields are parallel when

$$\boldsymbol{V} \begin{bmatrix} s \\ t \\ 1 \end{bmatrix} = \lambda \boldsymbol{W} \begin{bmatrix} s \\ t \\ 1 \end{bmatrix} \tag{10}$$

If $\boldsymbol{W}$ is invertible, multiplying this by its inverse, $\boldsymbol{W}^{-1}$, leads to

$$\boldsymbol{W}^{-1}\boldsymbol{V}\begin{bmatrix} s \\ t \\ 1 \end{bmatrix} = \lambda \begin{bmatrix} s \\ t \\ 1 \end{bmatrix} \tag{11}$$

which is an eigenvector problem $\boldsymbol{M}\boldsymbol{x} = \lambda\boldsymbol{x}$ with $\boldsymbol{M} = \boldsymbol{W}^{-1}\boldsymbol{V}$. If $\boldsymbol{W}$ is not invertible but $\boldsymbol{V}$ is, then the rôles of $\boldsymbol{v}$ and $\boldsymbol{w}$ can be swapped. If both are singular, no isolated solutions exist. Points where two linear vector fields are parallel can thus be found analytically by calculating eigenvectors of a 3 by 3 matrix $\boldsymbol{M}$. Points lying outside the triangle must be discarded.

A curve-following algorithm

Careful implementation of the three above methods guarantees *closed* lines, but a situation as in Fig. 4a may occur. Even if such a result is correct for the multilinearly interpolated discrete data, a more consistent set of lines with a different topology may be obtained.

This is why, in practice, techniques as in [1] or [18] have been shown to be successful: Starting from a seed point lying on the curve $\boldsymbol{v} \parallel \boldsymbol{w}$, the curve is constructed by stepping along it. Basically, a small step is first done along $\boldsymbol{v}$ and then corrected within the plane $N_{\boldsymbol{x},\boldsymbol{v}}$ for minimizing the magnitude of the projection of $\boldsymbol{w}$. Seed points can be computed by one of the preceding methods, or alternatively, the procedure may start at arbitrary points such as grid nodes. This variant (used also by [1]) works because correction steps must be bounded in size anyway.

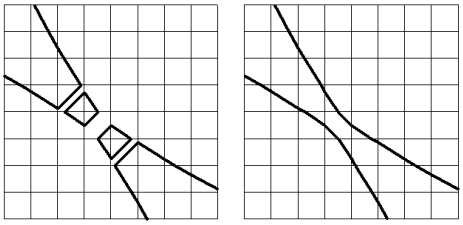

Figure 4: a) Possible and b) desired connectivity for the same data set.

6.2 Computing eigenvalues

Although eq. (3) and (5) are of the form $\boldsymbol{v} \parallel \boldsymbol{M}\boldsymbol{v}$ and thus have eigenvectors as their solutions, these equations are *not* eigenvalue problems. The difference is that both the matrix and the vector are known functions of space and the actual unknowns are the spatial coordinates $\boldsymbol{x}$. Therefore it is not necessary to calculate eigenvalues and eigenvectors of the matrix at specific locations, e.g. the grid nodes. If an algorithm for solving $\boldsymbol{v} \parallel \boldsymbol{w}$ is available, this is sufficient to solve eq. (3) or (5).

Even though eigenvalues are not needed for extracting lines, they can be required for the selection criterion. However, at this stage, a much smaller number of eigenvalues is required. And if only the one eigenvalue associated with $\boldsymbol{v}$ is needed, it can be cheaply computed by taking the ratio of non-zero components of $\boldsymbol{v}$ and $\boldsymbol{M}\boldsymbol{v}$.

7 Conclusions

We proposed the "parallel vectors" operator as a basic tool for vector field visualization, and we demonstrated that many existing techniques for extracting feature lines from field data can be expressed with this operator. Fig. 5 puts these methods into context.

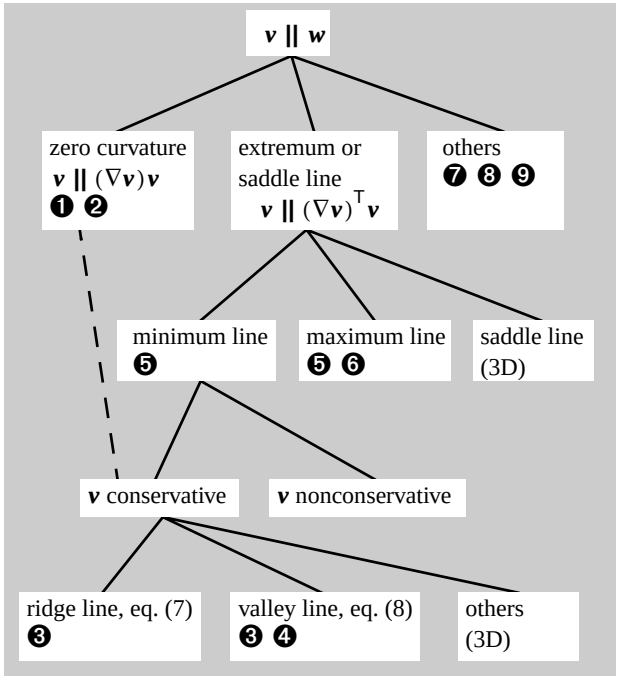

Figure 5: A tree of problems that can be expressed with the "parallel vectors" operator. ❶ through ❾ refer to the applications discussed in section 4.

We see the main benefit of our formalism in its decoupling of definition and algorithm. This way it can help to improve communication about feature extraction methods.

The line extraction procedure now has a fixed structure:

- Compute $\boldsymbol{v}$ and $\boldsymbol{w}$
- Compute the lines $\boldsymbol{v} \parallel \boldsymbol{w}$
- Apply selection criteria

We do not claim that this generic algorithm can beat, in quality or performance, the specific implementations given in the literature. The advantage of the modular approach is that code can be reused for new applications. It could also be used as a quick way to generate a second implementation for verification.

For these reasons, we believe that the "parallel vectors" operator would be worth including into future visualization environments.

For the AVS 5 environment, we implemented a "parallel vectors" plus a few auxiliary modules which can be downloaded from `http://www.cg.inf.ethz.ch/~rothma/turbo/parallel.html`

Acknowledgments

This work was supported by Sulzer Hydro Ltd. and by the Swiss Commission for Technology and Innovation. We thank Philippe de Forcrand, Markus Gross, Christian Mielke and Wesley Petersen for helpful comments and fruitful discussions.

References

[1] D.C. BANKS and B.A. SINGER, "Vortex Tubes in Turbulent Flows: Identification, Representation, Reconstruction", in *Proceedings of Visualization* '94, pp. 132-139, Oct. 1994.

[2] A.G. BELYAEV, I.A. BOGAEVSKI and T.L. KUNII, "Ridges and ravines on a surface and segmentation of range images", in *Vision Geometry VI, Proceedings of SPIE*, vol. 3168, pp. 106-114, 1977.

[3] M. BRÄNDLI, "Modelle und Algorithmen für die Extraktion geomorphologischer und hydrologischer Objekte aus digitalen Geländemodellen", *Diss. Univ. of Zürich*, 1997.

[4] BRETON DE CHAMP, "Note sur les lignes de faîte et de thalweg", *C.R.*, 39, pp. 647ff, 1854. (according to [21]).

[5] DE SAINT-VENANT, "Surfaces à plus grande pente constituées sur des lignes courbes", *Bulletin de la soc. philomath. de Paris*, 6 mars 1852. (according to [21]).

[6] R. FINSTERWALDER, "Zur Bestimmung von Tal- und Kammlinien", *Zeitschrift für Vermessungswesen*, 111(5), pp. 184-189, 1986.

[7] R.M. HARALICK, "Ridges and Valleys on Digital Images", *Computer Vision, Graphics, and Image Processing*, pp. 28-38, 1983.

[8] R.M. HARALICK and L.G. SHAPIRO, *Computer and Robot Vision*, vol. I, Addison-Wesley, 1992.

[9] L. HESSELINK and T. DELMARCELLE, "Visualization of vector and tensor data sets". In *Scientific visualization - Advances and challenges*, L. Rosenblum et al., Eds. Academic Press, pp. 367--390, 1994.

[10] J. JEONG and F. HUSSAIN, "On the identification of a vortex", in *Journal of Fluid Mech.*, vol. 285, pp. 69-94, 1995.

[11] V. INTERRANTE, H. FUCHS and S. PIZER, "Enhancing Transparent Skin Surfaces with Ridge and Valley Lines", in *Proceedings of Visualization* '95, pp. 52-59, 1995.

[12] E.G. JOHNSTON and A. ROSENFELD, "Digital detection of pits, peaks, ridges, and ravines". *IEEE Transactions on Systems, Man and Cybernetics*, SMC-5(4), pp. 472-480, 1975.

[13] D. KENWRIGHT, "Automatic Detection of Open and Closed Separation and Attachment Lines", in *Proceedings of Visualization '98*, pp. 151-158, 1998.

[14] Y. LEVY, D. DEGANI and A. SEGINER, "Graphical Visualization of Vortical Flows by Means of Helicity". *AIAA 28(8)*, pp. 1347-1352, August 1990.

[15] K.-L. MA and V. INTERRANTE, "Extracting Feature Lines from 3D Unstructured Grids", in *Proceedings of Visualization* '97, pp. 285-292, 1997.

[16] D. M. MARK, "Topology of ridge patterns: Randomness and constraints", Geological Society of America Bulletin, Part I, 90, pp. 164-172, 1979.

[17] C. MIELKE and L. KLEISER, "Investigation of the Late Stages of Transition to Turbulence in a Mach 4.5 Boundary Layer", in *Notes on Numerical Fluid Mechanics*, vol. 64, Vieweg Verlag, 1998.

[18] H. MIURA and S. KIDA, "Identification of Tubular Vortices in Turbulence", *Journal of the Physical Society of Japan*, vol. 66, nr. 5, pp. 1331-1334, May 1997.

[19] M. ROTH and R. PEIKERT, "Flow visualization for turbomachinery design", in *Proceedings of Visualization '96*, pp. 381-384, Oct. 1996.

[20] M. ROTH and R. PEIKERT, "A Higher-Order Method for Finding Vortex Core Lines", in *Proceedings of Visualization* '98, pp. 143-150, Oct. 1998.

[21] R. ROTHE, "Zum Problem des Talwegs", *Sitzungsberichte der Berliner mathematischen Gesellschaft*, vol. 14, pp. 51-68, 1915.

[22] D. SILVER and X. WANG, "Volume Tracking", in *Proceedings of Visualization* '96, pp. 157-164, 1996.

[23] R.C. STRAWN, D. KENWRIGHT and J. AHMAD, "Computer Visualization of Vortex Wake Systems", *American Helicopter Society 54th Annual Forum*, Washington DC, May 1998.

[24] D. SUJUDI and R. HAIMES, "Identification of Swirling Flow in 3D Vector Fields", *Tech. Report, Dept. of Aeronautics and Astronautics*, MIT, Cambridge, MA, 1995.

[25] C.-K. TANG and G. MEDIONI, "Extremal Feature Extraction from 3-D Vector and Noisy Scalar Fields", *Proceedings of Visualization* '98, pp. 95-102, 1998.

[26] J.-P. THIRION and A. GOURDON, "The 3D Marching Lines Algorithm", *Graphical Models and Image Processing*, vol. 58, no. 6, pp. 503-509, 1996.

[27] C. WERNER, "Formal Analysis of Ridge and Channel Patterns in Maturely Eroded Terrain", Annals of the Association of American Geographers, 78(2), pp. 253-270, 1988.

[28] R. WESTERMANN and T. ERTL, "A Multiscale Approach to Integrated Volume Segmentation and Rendering", in *Proceedings of Eurographics* '98, vol. 17, nr. 3, 1998.

[29] C. WIENER, *Lehrbuch der darstellenden Geometrie*, vol. 2, Leipzig 1887. (according to [6]).

Appeared in *Proceedings of IEEE Visualization '99*, pp. 263-270, 1999.