

# Detection and Visualization of Vortices

Ming Jiang\*      Raghu Machiraju\*      David Thompson†

\*Department of Computer and Information Science  
The Ohio State University

†Department of Aerospace Engineering  
Mississippi State University

## 1 Introduction

In general, a feature can be defined as a pattern occurring in a dataset that is the manifestation of correlations among various components of the data. For many features that occur in scientific data, these correlations can be defined precisely. For other features, they are not well understood or do not lend themselves to precise definitions. Surprisingly, the swirling feature in flow fields, commonly referred to as a vortex, is an example of a feature for which a precise definition does not exist.

By most accounts [1–3], a vortex is characterized by the swirling motion of fluid around a central region. This characterization stems from our visual perception of swirling phenomena that are pervasive throughout the natural world. However, translating this intuitive description of a vortex into a formal definition has been quite a challenge.

Lugt [1] proposed the following definition for a vortex: *A vortex is the rotating motion of a multitude of material particles around a common center.* The problem with this definition is that it is too vague. Although it is consistent with visual observations, it does not lend itself readily to implementation in a detection algorithm. In light of this, Robinson [3] attempted to provide a more concrete definition of a vortex by specifying the conditions for detecting swirling flows in three dimensions:

*A vortex exists when instantaneous streamlines mapped onto a plane normal to the vortex core exhibit a roughly circular or spiral pattern, when viewed from a reference frame moving with the center of the vortex core.*

The primary shortcoming of this operational definition is that it is self-referential: the existence of a vortex requires *a priori* knowledge of the orientation and motion of its core.

Despite the lack of a formal definition, various detection algorithms have been implemented that can adequately identify vortices in most computational datasets. In this paper, we present an overview of existing detection methods; in particular, we focus on nine methods that are representative of the state-of-the-art. Although this is not a complete listing of vortex detection algorithms, the range of relevant issues covered by these nine methods is comprehensive in scope. The methods are:

- *Helicity* Method by Levy et al. [4]
- *Swirl Parameter* Method by Berdahl and Thompson [5]
- *Lambda<sub>2</sub>* Method by Jeong and Hussain [6]
- *Predictor-Corrector* Method by Banks and Singer [7]
- *Eigenvector* Method by Sujudi and Haines [8]
- *Parallel Vectors* Method by Roth and Peikert [9]
- *Maximum Vorticity* Method by Strawn et al. [10]
- *Streamline* Methods by Sadarjoen et al. [11]
- *Combinatorial* Method by Jiang et al. [12]

We first present three taxonomies for classifying these nine detection methods in Section 2. We then describe each algorithm in Section 3, along with pseudocode where appropriate. Next, we describe a recently developed verification algorithm for swirling flows in Section 4. In Section 5, we discuss the different visualization techniques for vortices. Finally, we conclude and highlight future directions in this field.

## 2 Taxonomy

Almost every paper published on the subject of vortex detection has presented a classification of its predecessors in some fashion. One of the most comprehensive classifications of vortex detection methods was proposed by Roth in [13]. In this section, we present three taxonomies for classifying existing detection methods. These taxonomies are based on how the vortex

Method	Region/Line	Galilean	Local/Global
Helicity	Line	Not Invariant	Local
Swirl Parameter	Region	Not Invariant	Local
$\Lambda_2$	Region	Invariant	Local
Predictor-Corrector	Line	Invariant	Global
Eigenvector	Line	Not Invariant	Local
Parallel Vectors	Line	Not Invariant	Local
Maximum Vorticity	Line	Invariant	Local
Streamline	Region	Not Invariant	Global
Combinatorial	Region	Not Invariant	Local

Table 1: Taxonomies of vortex detection algorithms.

is defined, whether or not the detection method is Galilean invariant, and the local or global nature of the identification process.

The first taxonomy classifies detection methods based on the definition of a vortex. A vortex can be defined either as a region or as a line. A region-based vortex definition specifies criteria for identifying contiguous grid nodes (or cells) that belong to either the vortex or its core. A line-based vortex definition, on the other hand, specifies criteria for locating vortex core lines. A set of contiguous line segments constitutes the vortex core line. In general, detection algorithms corresponding to region-based definitions are easier to implement and computationally cheaper than their line-based counterparts. Line-based algorithms must precisely locate points where the vortex core line intersect the grid cells. However, line-based algorithms provide more compact representations of vortices and can easily distinguish between individual vortices in close proximity. The latter is problematic for region-based approaches. Table 1(column 1) categorizes the nine detection methods based on this criterion.

The second taxonomy classifies detection methods based on whether or not they are Galilean (Lagrangian) invariant. Most detection methods work under the assumption of either steady flow fields or vortices moving much slower than the average fluid particle. In a time-varying flow field, a vortex exhibits swirling motion only when viewed from a reference frame that moves with the vortex [1,3]. In order to detect vortices in unsteady (time-dependent) flows, it is necessary for the method to satisfy Galilean invariance. A detection method is Galilean invariant if it produces the same results when a uniform velocity is added to the existing velocity field. Thus, methods which do not depend directly on the velocity, such as pressure or

vorticity, are Galilean invariant. This is an important property especially in the context of tracking vortices in time-varying flow fields. Table 1(column 2) categorizes the nine detection methods based on this criterion.

The third taxonomy classifies detection methods based on the local or global nature of the identification process. A detection method is considered to be local if the identification process requires only operations within the local neighborhood of a grid cell. Methods that rely on the velocity gradient tensor are usually local methods. On the other hand, a global method requires examining many grid cells in order to identify vortices. Methods that involve tracing streamlines in velocity or vorticity fields are considered to be global. From the definitions in the preceding section, it is apparent that a vortex is a global feature. It may be preferable to detect global features using global methods; however, on the basis of computation, global detection methods tend to be more expensive than local methods. However, in order to verify the accuracy of the detected results, a global approach is necessary. We describe this aspect in more detail in Section 4. Table 1(column 3) categorizes the nine detection methods based on this criterion.

### 3 Vortex Detection Algorithms

#### 3.1 Helicity Method

Levy et al. [4] introduced the use of normalized helicity  $H_n$  for extracting vortex core lines, though they were not the first to identify the strong correlation between helicity and coherent structures in turbulent flow fields.  $H_n$  is a scalar quantity defined everywhere except at critical points:

$$H_n = \frac{\mathbf{v} \cdot \boldsymbol{\omega}}{|\mathbf{v}||\boldsymbol{\omega}|} \quad (1)$$

$H_n$  is the cosine of the angle between velocity  $\mathbf{v}$  and vorticity  $\boldsymbol{\omega}$ . The underlying assumption is that near vortex core regions, the angle between  $\mathbf{v}$  and  $\boldsymbol{\omega}$  is small. In the limiting case, where  $\mathbf{v} \parallel \boldsymbol{\omega}$ ,  $H_n = \pm 1$ , and the streamline that passes through that point has zero curvature (straight line). The authors suggested an approach to extract vortex core lines by first locating maximal points of  $H_n$  on cross sectional planes, which are also points of minimal streamline curvature, and then growing the core line by tracing a streamline from the maximal points.

The sign of  $H_n$  indicates the direction of swirl (clockwise or counterclockwise) of the vortex with respect to the streamwise velocity component. It

switches whenever a transition occurs between primary and secondary vortex. The authors successfully used this feature with corresponding colors to distinguish between the primary and secondary vortices in the hemisphere-cylinder and ogive-cylinder datasets. However, the extracted core line may not always correspond to the actual vortex core line [13].

### 3.2 Swirl Parameter Method

Berdahl and Thompson [5] presented a vortex detection method based on the connection between swirling motion and the existence of complex eigenvalues in the velocity gradient tensor  $\mathbf{J}$ . The authors introduced the intrinsic swirl parameter  $\tau$ , defined by the ratio of the convection time  $t_{conv}$  (the time for a fluid particle to convect through the region of complex eigenvalues  $R_C$ ) to the orbit time  $t_{orbit}$  (the time for a fluid particle to return to the same angular position). Thus,

$$t_{conv} = \frac{2\pi}{|Im(\lambda_C)|} \quad t_{orbit} = \frac{L}{|\mathbf{v}_{conv}|} \quad (2)$$

where  $Im(\lambda_C)$  is the imaginary part of the complex conjugate pair of eigenvalues,  $L$  is the characteristic length associated with the size of  $R_C$ , and  $\mathbf{v}_{conv}$  is the convection velocity aligned along  $L$ . From Equation 2,  $\tau$  can be written as:

$$\tau = \frac{t_{conv}}{t_{orbit}} = \frac{|Im(\lambda_C)|L}{2\pi|\mathbf{v}_{conv}|} \quad (3)$$

When  $\tau \rightarrow 0$ , the fluid particle convects too rapidly through  $R_C$  to be “captured” by the vortex. Thus  $\tau$  is nonzero in regions containing vortices and attains a local maximum in the vortex core. For three dimensions, the length and orientation of  $L$  are unknown, because in general there is no single plane of swirling flow. The authors suggest using the plane normal to either the vorticity vector  $\boldsymbol{\omega}$  or the real eigenvector  $\mathbf{e}_R$ , which are local approximations to the actual vortex core direction vector. The convective velocity  $\mathbf{v}_{conv}$  is computed by projecting the local velocity vectors onto this plane:

$$\mathbf{v}_{conv} = \mathbf{v} - (\mathbf{v} \cdot \mathbf{n})\mathbf{n} \quad (4)$$

where  $\mathbf{n}$  is the plane normal computed from either  $\boldsymbol{\omega}$  or  $\mathbf{e}_R$ .

Figure 1 (courtesy of Michael Remotigue, Mississippi State University) illustrates the results when this method is applied to the propeller dataset. In the left image, the intensity of  $\tau$  is described by a colormap. In the right image, isosurfaces are generated showing the path of the tip vortex as well

as a ring vortex that was shed off the propeller base. However, selecting the right threshold for  $\tau$  in order to distinguish individual vortices is often difficult.

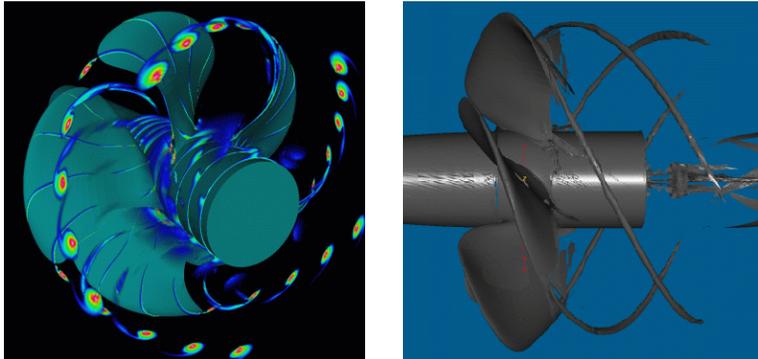


Figure 1: Swirl parameter

### 3.3 $\Lambda_2$ Method

Jeong and Hussain [6] proposed a definition for a vortex that is commonly referred to as the  $\lambda_2$ -definition. They begin with the premise that a pressure minimum is not sufficient as a detection criterion. The problems are due to unsteady irrotational straining, which can create a pressure minimum in the absence of a vortex, and viscous effects, which can eliminate the pressure minimum within a vortex. To remove these effects, they decompose the velocity gradient tensor  $\mathbf{J}$  into its symmetric part, the rate of deformation or strain-rate tensor  $\mathbf{S}$ , and antisymmetric part, the spin tensor  $\mathbf{\Omega}$ , and consider only the contribution from  $\mathbf{S}^2 + \mathbf{\Omega}^2$ .

$$\mathbf{S} = \frac{\mathbf{J} + \mathbf{J}^T}{2} \quad \mathbf{\Omega} = \frac{\mathbf{J} - \mathbf{J}^T}{2} \quad (5)$$

They define a vortex as a connected region where  $\mathbf{S}^2 + \mathbf{\Omega}^2$  has two negative eigenvalues. Because  $\mathbf{S}^2 + \mathbf{\Omega}^2$  is real and symmetric, it has only real eigenvalues. Let  $\lambda_1$ ,  $\lambda_2$ , and  $\lambda_3$  be the eigenvalues such that  $\lambda_1 \geq \lambda_2 \geq \lambda_3$ . If  $\lambda_2$  is negative at a point, then that point belongs to a vortex core. Through several analytical examples and direct numerical simulation datasets, the authors demonstrated the effectiveness of the  $\lambda_2$ -definition compared to others. However, in situations where several vortices exist, it can be difficult for this method to distinguish between individual vortices.

### 3.4 Predictor-Corrector Method

The vorticity-predictor, pressure-corrector method for detecting vortex core lines was proposed by Banks and Singer [7,14]. Their underlying assumption is that vortical motion is sustained by pressure gradients and indicated by vorticity  $\omega$ . The algorithm extracts a skeleton approximation to the vortex core by tracing vorticity lines and then correcting the prediction based on local pressure minimum. In order to find the initial set of seed points for tracing vorticity lines, they consider grid points with low pressure and high vorticity magnitude. However, as the authors pointed out, it is possible for a grid point to satisfy both conditions without being part of a vortex core. An outline of the algorithm is provided in Algorithm 1.

```
1: locate seed points with low pressure and high  $|\omega|$ 
2: for all seed points do
3:   repeat
4:     compute  $\omega_i$  at current skeleton point
5:     step in  $\omega_i$  direction to predict next point
6:     compute  $\omega_{i+1}$  at predicted point  $\mathbf{P}_\omega$ 
7:     locate minimum pressure  $\mathbf{P}_p$  on plane  $\perp \omega$ 
8:     if  $dist(\mathbf{P}_\omega, \mathbf{P}_p) < \text{threshold}$  then
9:       correct next point to  $\mathbf{P}_p$ 
10:    else
11:      terminate skeleton growth
12:    end if
13:    eliminate seed points within distance $_r$ 
14:  until skeleton exits domain or is too long
15: end for
```

Algorithm 1: Predictor-corrector method

For the predictor step, vorticity integration can be performed using fourth-order Runge-Kutta. The authors, instead, suggested a simplification whereby the step size corresponds to the smallest dimension of the local grid cell. For the corrector step, steepest descent is used to find the local pressure minimum, with the step size, again, being the smallest grid cell dimension.

Algorithm 1 terminates when the minimum pressure point is too far from the predicted point; however, the method is not guaranteed to terminate in every case, because the growing skeleton can form closed loops, which is not ideal for real vortices. Furthermore, special care must to be taken in order to minimize the number of skeletons approximating the same vortex core

line, since the skeleton grown from each seed point may end up describing the same vortex core.

### 3.5 Eigenvector Method

The eigenvector method for detecting vortex core lines was first proposed by Sujudi and Haines [8]. The method is based on critical-point theory, which asserts that the eigenvalues and eigenvectors of the velocity gradient tensor  $\mathbf{J}$ , evaluated at a critical point, define the local flow pattern about that point. As the authors pointed out, there are swirling flows which do not contain critical points within its center. In order to handle these cases, velocity vectors are projected onto the plane normal to the eigenvector of the real eigenvalue, assuming the other two eigenvalues are complex conjugate pairs, to see if they are zero. If they are, then the point must be part of vortex core. An outline of the algorithm is given in Algorithm 2.

```

1: decompose grid cells into tetrahedral cells
2: for all tetrahedral cells do
3:   linearly interpolate  $\mathbf{v}$  to produce  $\mathbf{J}$ 
4:   compute all three eigenvalues of  $\mathbf{J}$ 
5:   if two eigenvalues are complex conjugates then
6:     compute eigenvector  $\mathbf{e}_R$  for the real eigenvalue
7:     project  $\mathbf{v}$  onto  $\mathbf{e}_R \rightarrow$  reduced velocity  $\mathbf{v}_r$ 
8:     compute the zero  $\mathbf{v}_r$  straight line  $\gamma_z$ 
9:     if  $\gamma_z$  intersects cell twice then
10:      add line segment to vortex core
11:    end if
12:  end if
13: end for

```

Algorithm 2: Eigenvector method

Initially, all mesh elements are decomposed into tetrahedral cells. Linear interpolation of  $\mathbf{v}$  within the cell follows, which induces a constant  $\mathbf{J}$ . The reduced velocity  $\mathbf{v}_r$  is computed by subtracting the velocity component in the direction of  $\mathbf{e}_R$ , and is equivalent to projecting  $\mathbf{v}$  onto the plane normal to  $\mathbf{e}_R$ . Finding the zero locations on the plane requires setting up a system of three equations using the linearly interpolated components of  $\mathbf{v}_r$ , which can be solved using any two of the three linearly independent equations. The solution is a straight line of zero  $\mathbf{v}_r$ .

This method was successfully applied to detecting vortex cores in numerous CFD applications [15, 16]. Figure 2 (courtesy of Robert Haimes, Massachusetts Institute of Technology) illustrates one such example taken from [16]. The yellow line segments represent the vortex cores extracted from a transient F/A-18 simulation dataset. However, as the authors pointed out, producing contiguous vortex core lines is not always possible because the underlying interpolant may not be linear or line segments may not meet up at shared faces. Modifications to the original algorithm are proposed in [17] to address this issue and improve its performance.

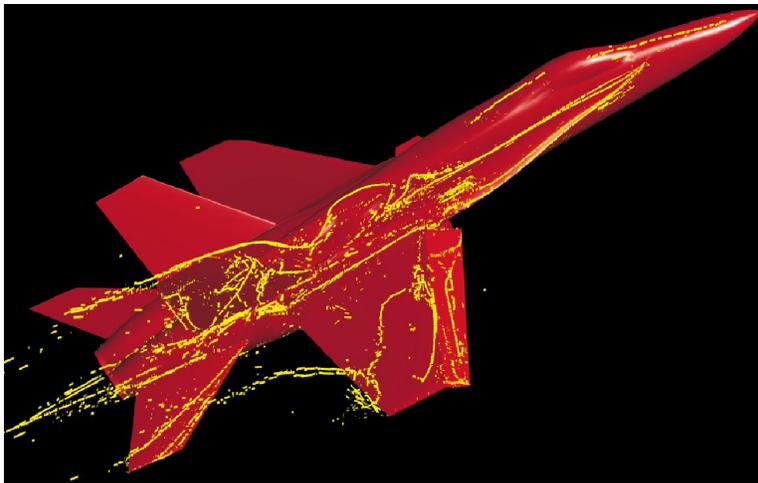


Figure 2: Eigenvector approach ((©1998 IEEE))

### 3.6 Parallel Vectors Method

The parallel vectors operator was first introduced by Roth and Peikert [9] as a higher-order method for locating vortex core lines. They recast the first-order eigenvector method into a parallel alignment problem between  $\mathbf{v}$  and its first derivative  $\mathbf{J}\mathbf{v}$  (i.e., reduced velocity is zero when  $\mathbf{v}$  is parallel to the real eigenvector of  $\mathbf{J}$ ). In order to better capture slowly rotating curved vortices that are typical in turbomachinery flow fields, they use the second derivative of  $\mathbf{v}$  which is defined as:

$$\mathbf{w} = \frac{D^2\mathbf{v}}{Dt^2} = \frac{D(\mathbf{J}\mathbf{v})}{Dt} = \mathbf{J}\mathbf{J}\mathbf{v} + \mathbf{T}\mathbf{v}\mathbf{v} \quad (6)$$

where  $\mathbf{T}$  is a  $3 \times 3 \times 3$  tensor. Essentially, a vortex core line is the locus where  $\mathbf{v}$  is parallel to  $\mathbf{w}$ :  $\{x : \mathbf{v}(x) \times \mathbf{w}(x) = 0\}$ . An outline of the algorithm is

given in Algorithm 3.

```

1: for all grid points do
2:   calculate  $\mathbf{J}$  and compute  $\mathbf{v}' = \mathbf{J}\mathbf{v}$ 
3:   calculate  $\mathbf{J}'$  and compute  $\mathbf{w} = \mathbf{J}\mathbf{v}'$ 
4: end for
5: for all grid faces do
6:   find zero of function  $\mathbf{v} \times \mathbf{w}$ 
7:   use Newton iterations starting from face center
8:   if zero lies on face then
9:     connect with straight line to previous zero
10:  end if
11: end for

```

Algorithm 3: Parallel vectors method

Due to discretization errors, excessive fluctuations may result from computing the higher-order derivatives. To avoid this, the authors recommend smoothing the vector field data as a preprocessing step. In [13, 18], other approaches for finding parallel vectors are presented, along with *post priori* criteria for removing line segments that might be of insufficient strength (speed of local rotation) or quality (angle between velocity at core and core line).

Figure 3 (courtesy of Martin Roth, Swiss Federal Institute of Technology Zürich) illustrates the results for the Francis turbine runner dataset and the stator of a reversible pump-turbine dataset. The black line segments indicate the locations of detected vortex core lines. Note the existence of gaps in the detected core lines, which are mainly due to the large number of raw solution lines produced by the higher-order method [13].

### 3.7 Maximum Vorticity Method

Strawn et al. [10] define a vortex core as a local maximum of vorticity magnitude  $|\boldsymbol{\omega}|$  in the plane normal to  $\boldsymbol{\omega}$ . This technique is applicable for free-shear flows, but not shear layers, which have high  $|\boldsymbol{\omega}|$  but no local  $|\boldsymbol{\omega}|$  maxima. The motivation for this approach comes from situations where multiple vortices with the same orientation and overlapping cores are in close proximity. The resulting velocity field would only exhibit a single rotational center. To address this issue, the authors introduced the maximum vorticity method outlined in Algorithm 4.

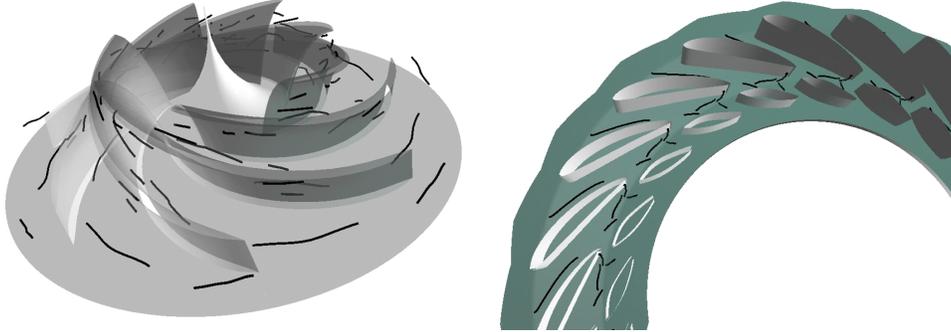


Figure 3: Parallel vector operator

For the preprocessing step,  $\omega$  is transformed into computational space, where the search for  $|\omega|$  maxima is done on a uniform grid. The gradient of  $|\omega|$  is assumed to vary bilinearly over the grid face. Finding the solution points where  $\nabla|\omega| = 0$  requires solving a pair of quadratic equations derived from the bilinear interpolation function. The authors also suggest using two thresholds to eliminate some of the weaker vortex centers. The first threshold eliminates cell faces with low  $|\omega|$ , and the second threshold eliminates cell faces whose normal may be misaligned with  $\omega$ . This method was successfully applied to distinguish individual vortices in the delta wing dataset (primary, secondary, and tertiary vortices) and the V-22 tiltrotor blades dataset (tip and root vortices from each rotor blade).

### 3.8 Streamline Methods

Sadarjoen et al. [11] proposed an efficient algorithm for detecting vortices using the winding angle method. The winding angle concept was first proposed by Portela [2] in a mathematically rigorous but computationally expensive fashion. Essentially, given a two-dimensional streamline, the winding angle measures the amount of rotation of the streamline with respect to a point. Sadarjoen et al. [11,19,20] simplified the definition and proposed an efficient algorithm for extracting two-dimensional vortices based on it. By their definition, the winding angle  $\alpha_w$  of a streamline is a measure of the cumulative change of direction of streamline segments.

$$\alpha_w = \sum_{i=1}^{N-2} \angle(\mathbf{p}_{i-1}, \mathbf{p}_i, \mathbf{p}_{i+1}) \quad (7)$$

```

1: compute  $\omega$  at all grid nodes
2: for all cell faces do
3:   examine its  $4 \times 4$  surrounding nodes
4:   if  $\exists$  maximum  $|\omega|$  in central nodes then
5:     mark grid face as candidate face
6:   end if
7: end for
8: for all candidate faces do
9:   compute  $\nabla|\omega|$  using central difference at nodes
10:  compute solution points where  $\nabla|\omega| = 0$ 
11:  if points within face and are local maxima then
12:    mark them as vortex core points
13:  end if
14: end for

```

Algorithm 4: Maximum vorticity method

where  $\mathbf{p}_i$  are the  $N$  streampoints of the streamline, and  $\angle(\mathbf{p}_{i-1}, \mathbf{p}_i, \mathbf{p}_{i+1})$  measures the signed angle between the two line segments delimited by  $\mathbf{p}_{i-1}$ ,  $\mathbf{p}_i$ , and  $\mathbf{p}_{i+1}$ , with counterclockwise rotation being positive and clockwise rotation being negative. Therefore, a vortex exists in a region where  $\alpha_w \geq 2\pi$  for at least one streamline. For slowly rotating vortices, the  $2\pi$  winding criterion can be relaxed appropriately. An outline of the method is given in Algorithm 5.

Once the winding streamlines are marked, a clustering algorithm, based on the distance between center point and cluster, is used to group the streamlines that belong to the same vortex. The location of each cluster is taken to be the location of the vortex core. Various attributes of the vortex, such as shape and orientation are used to quantitatively visualize the vortices. Figure 4 (courtesy of I. Ari Sadarjoen, Delft University of Technology) depicts the results when the method is applied to a slice of the tapered cylinder dataset. Elliptical icons are used to represent the shape of the extracted vortices, and the two colors (green and red) are used to represent the two different orientations.

Yet another streamline method is the curvature density center method for locating vortex cores in two-dimensional flow fields [11, 19, 20]. Pagen-darm et al. [21] extended the method for three-dimensional flow fields. The underlying assumption behind this approach is that the center of curvature for each point on a winding streamline should form a tight cluster, and

```

1: select an initial set of seed points
2: for all seed points do
3:   trace its streamline and compute  $\alpha_w$ 
4:   if  $|\alpha_w| \geq 2\pi$  and initial point near end point then
5:     mark streamline as winding
6:   end if
7: end for
8: for all winding streamlines do
9:   compute its center point  $\mathbf{c}$  (geometric mean)
10:  if  $\mathbf{c} \notin$  vortex clusters then
11:    add  $\mathbf{c}$  to vortex clusters
12:  end if
13: end for

```

Algorithm 5: Winding angle method

the local maxima within this cluster is the vortex core. By computing the curvature center at each sample point throughout the domain, a density field is formed whose peaks are the locations of vortex cores. As pointed out in [11,19,20], this approach lacks the robustness to work well for non-circular flows, such as the elliptically shaped vortices illustrated in Figure 4.

### 3.9 Combinatorial Method

Jiang et al. [12] presented a method for extracting vortex core regions based on ideas from combinatorial topology. In this approach, a combinatorial labeling scheme based on *Sperner's Lemma* is applied to the velocity vector field in order to identify centers of swirling flows. The origin of Sperner's Lemma lies in the *Fixed Point Theory* of combinatorial topology. The connection between vortices and fixed points (i.e., critical points) are well known [22,23]. Whereas Sperner's Lemma labels the vertices of a simplicial complex and identifies the fixed points of the labeled subdivision, the proposed method labels the velocity vectors at grid nodes and identifies grid cells that are most likely to contain critical points.

Each velocity vector  $\mathbf{v}$  is labeled according to the direction range in which it points. It is sufficient to examine the surrounding nodes of a grid cell for the existence of revolving velocity vectors. The number of direction ranges corresponds to the number of surrounding nodes. (For a quadrilateral mesh, there are four direction ranges, each spanning  $90^\circ$ . For two-dimensional flow fields, a grid cell belongs to a vortex core region if each of the four velocity

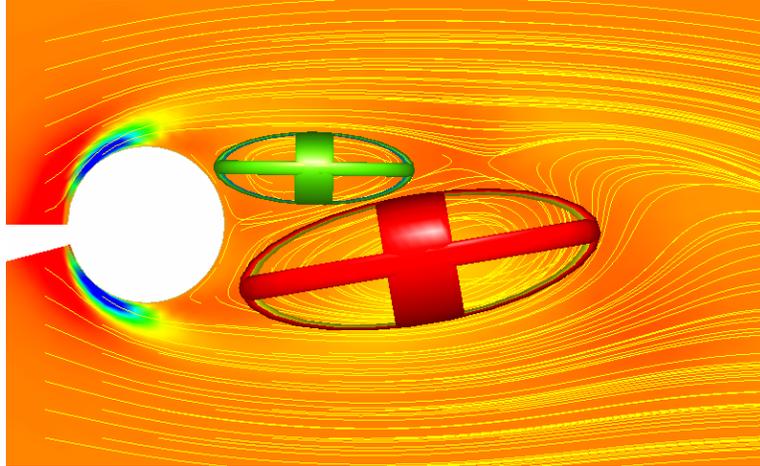


Figure 4: Winding angle method

vectors from the surrounding nodes point in a unique direction range, or satisfy the direction-spanning criterion. For three-dimensional flow fields, it is necessary to approximate the local swirling plane at each grid cell, and then project the surrounding velocity vectors onto this plane. An outline of the three-dimensional algorithm is given in Algorithm 6.

```

1: for all grid cells do
2:   compute swirl plane normal  $\mathbf{n}$  at cell center
3:   project  $\mathbf{v}$  from surrounding nodes
4:   for all  $\mathbf{v}_p$  in swirl plane do
5:     compute its angle  $\alpha$  from local x-axis
6:     label direction range for  $\alpha$ 
7:   end for
8:   if all direction ranges are labeled then
9:     mark grid cell as vortex core
10:  end if
11: end for

```

Algorithm 6: Combinatorial method

The authors use a simple region growth algorithm along with Algorithm 6 in order to segment the individual vortex core regions. What makes this method effective is its insensitivity to approximations to the local swirl plane normal  $\mathbf{n}$ . Figure 5 shows the results from this method on the the blunt fin

dataset. The yellow regions are detected vortex core regions, visualized using isosurfaces. The blue lines are the streamlines seeded near the detected vortex cores, and they serve to demonstrate the success of this approach by showing that the detected vortex cores actually lie in the center of the swirling flow. However, this approach can produce false positives [24].

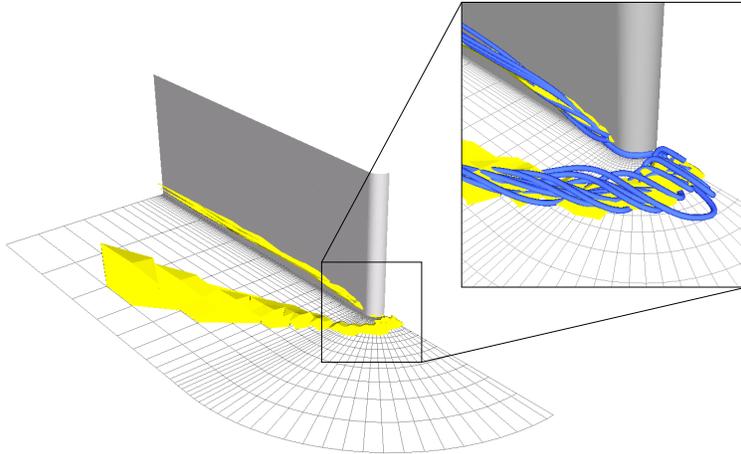


Figure 5: Combinatorial method (©2002 IEEE)

## 4 Swirling Flow Verification

The main deficiency common to all these detection algorithms is not the false positives which they may produce, but rather their inability to automatically distinguish between the false positives and the actual vortices. Imprecise vortex definitions or numerical artifacts are just two of the reasons why these false positives occur. The fundamental problem is that most detection algorithms employ local operators (e.g., velocity gradient tensor  $\mathbf{J}$ ) for detecting global features. As pointed out by Thompson et al. [25], these local operators are problematic because they do not incorporate the necessary global information into the detection process.

The most direct approach for verifying if a candidate feature is indeed a vortex is by visual inspection. The primary problem with this approach is that it requires human intervention, a process that is contrary to the automatic nature of the detection algorithms. The geometric verification algorithm proposed by Jiang et al. [24] addresses this issue by automating the verification process. By identifying the swirling streamlines surrounding

a candidate vortex core, the verification algorithm can arbitrate the presence or absence of a vortex most consistent with visual scrutiny.

As a post-processing step, the verification algorithm can work with any detection algorithm. Given a candidate vortex core, the goal is to identify the swirling streamlines surrounding it by using various differential geometry properties of the streamlines. The algorithm was designed for three-dimensional flow fields; in the two-dimensional case, using the winding angle method discussed in Section 3.8 to verify planar swirling streamlines is sufficient. Identifying three-dimensional swirling streamlines is non-trivial since vortices can bend and twist in various ways. An outline of the verification algorithm for a candidate vortex core is given in Algorithm 7.

```

1: uniformly distribute seed points at start position
2: for all seed points do
3:   for  $i = 0$  to  $N$  do
4:     trace next streampoint
5:     compute tangent vector  $\mathbf{t}$  and probe vector
6:     probe vortex core for swirl plane normal  $\mathbf{n}$ 
7:     align  $\mathbf{n}$  to z-axis and save transformation
8:     apply transformation to  $\mathbf{t} \rightarrow \mathbf{t}_a$ 
9:     project  $\mathbf{t}_a$  on (x,y)-plane  $\rightarrow \mathbf{t}_p$ 
10:    if  $\angle(\mathbf{t}_p^0, \mathbf{t}_p^i) \geq 2\pi$  then
11:      accept candidate vortex core
12:    end if
13:  end for
14: end for

```

Algorithm 7: Geometric verification algorithm

The verification algorithm begins by locating the upstream extent (tip) of the candidate vortex core. For candidate core lines, this is trivial; for candidate core regions, the authors in [24] proposed a bounding box heuristic. The initial position is the tip of the candidate vortex core. Seed points are distributed uniformly on a circle in the swirl plane at the start position. Once the projected tangent vectors makes a full revolution in the (x,y)-plane (i.e., satisfy the  $2\pi$  swirling criterion), the candidate vortex core is accepted as an actual vortex core.

Figure 6 depicts the results for the delta wing dataset. In the left image, the yellow regions are actual vortex cores and the green regions are false positives, artifacts from the combinatorial method. The middle image depicts

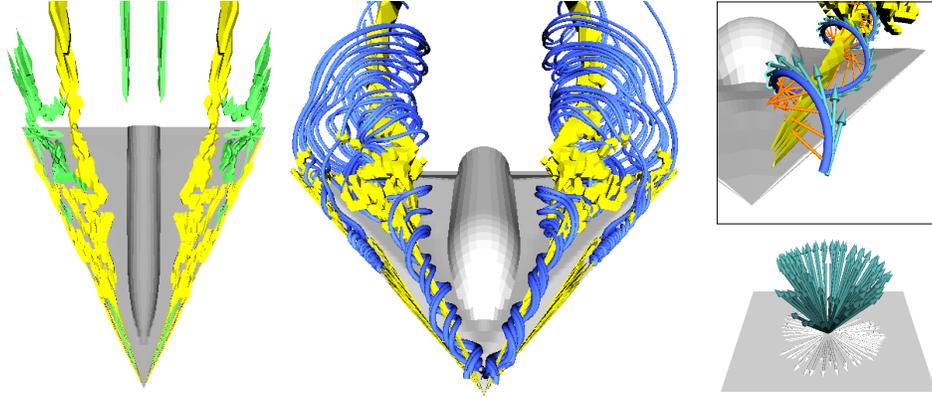


Figure 6: Geometric verification (©2002 IEEE)

the swirling streamlines surrounding the verified vortex cores. The right image shows the manner in which Algorithm 7 confirms that the identified candidate is indeed a vortex core. The cyan arrows represent the tangent vectors and the orange arrows represent the probe vectors. The bottom image on the right illustrates the projected tangent vectors revolving in the  $(x,y)$ -plane.

## 5 Visualization of Vortices

Methods used to visualize vortices are inextricably linked to the manner in which the vortices are detected. For example, line-based algorithms produce results that can best be visualized as line segments, as shown in Figures 2. In contrast, results generated by region-type algorithms can best be visualized using colormaps or isosurfaces, as shown in Figure 1. Additionally, iconic representations, such as the elliptical icons shown in Figure 4, can also be used to quantitatively visualize various attributes of vortices.

By seeding streamlines near vortex cores, the swirling patterns that are generally associated with vortices can be visualized. This is one of the primary techniques to ascertain the accuracy of detected results, either manually or automatically (see Section 4). Figure 7 illustrates how some of the pioneers in this field leverage this technique to validate or invalidate results from detection algorithms. The top left image (courtesy of I. Ari Sadarjoen, Delft University of Technology) illustrates the Pacific Ocean dataset where streamlines (cyan lines) are seeded throughout the domain to show regions of winding streamlines. The intent [11] was to demonstrate the inef-

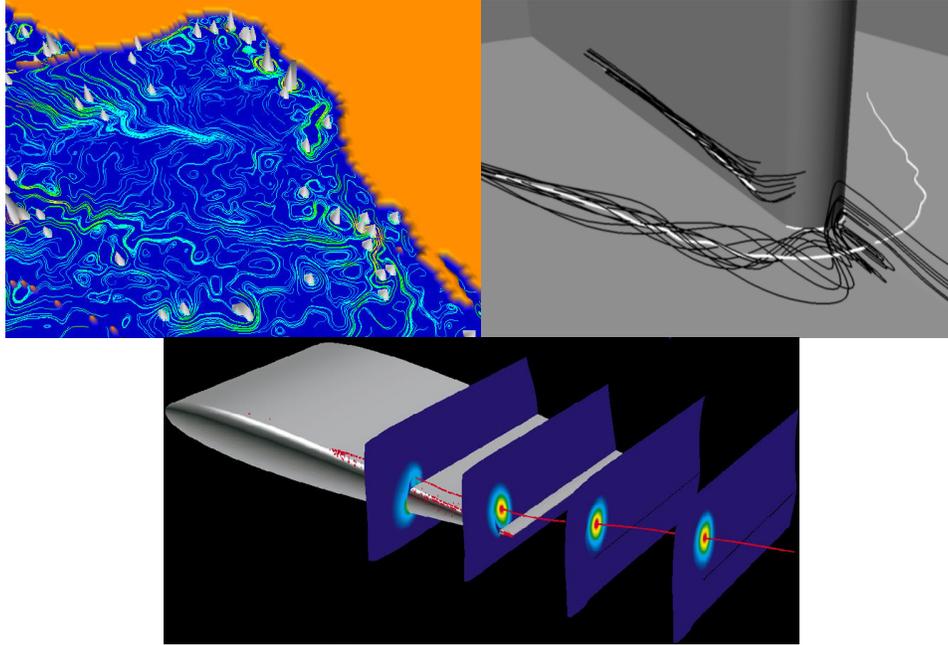


Figure 7: Visualization of vortices (©1998 IEEE)

fectiveness of the curvature center density method. The density peaks (gray isosurfaces) do not correspond well with the winding streamlines. The top right image (courtesy of Martin Roth, Swiss Federal Institute of Technology Zürich) depicts the vortical flow in the blunt fin dataset. Vortex core lines (white lines) were extracted using the parallel vectors method. In this case [9], the intent was to demonstrate the effectiveness of their method for extracting vortex core lines that correspond exactly to the center of swirling streamlines (black lines).

Besides seeding streamlines, the cutting plane technique is also preferred. Each cutting plane takes a sample slice of the dataset along a certain direction, and the visualization method can be isocontours of a scalar quantity or line-integral convolution (LIC) [26] of velocity vectors. The bottom image of Figure 7 depicts the wing-tip dataset where vortex core lines (red line segments) were extracted using the eigenvector method. Sample slices were taken [16] along the detected vortex core to demonstrate the correspondence between the isocontours and the extracted core line.

## 6 Conclusion

Throughout the past decade, there has been a steady stream of publications on the subject of vortex detection. We presented an overview of nine detection algorithms that are representative of the state-of-the-art. Each detection algorithm is classified based on how it defines a vortex, whether or not it is Galilean invariant, and the local or global nature of its identification process. Although many of the algorithms share similarities, each has its own advantages and disadvantages. A recently developed verification algorithm, that can be used in conjunction with any detection method, was also overviewed, as well as various techniques for visualizing detected vortices.

Although much progress has been made towards detecting vortices in steady flow fields, there is still a paucity of methods that can do the same in unsteady (time-varying) flow fields. None of the detection methods described in this paper can adequately address all of the issues unique to unsteady vortical flows. A major challenge will be to develop efficient and robust vortex detection and tracking algorithms for unsteady flow fields.

## 7 Acknowledgements

This work is partially funded by the National Science Foundation under the Large Data and Scientific Software Visualization Program (ACI-9982344), the Information Technology Research Program (ACS-0085969), an NSF Early Career Award (ACI-9734483), and a grant from the US Army Research Office (DAA-D19-00-1-0155).

## References

- [1] H. J. Lugt. *Vortex Flow in Nature and Technology*. Wiley, 1972.
- [2] L. M. Portela. *Identification and Characterization of Vortices in the Turbulent Boundary Layer*. PhD thesis, Stanford University, 1997.
- [3] S. K. Robinson. Coherent Motions in the Turbulent Boundary Layer. *Ann. Rev. Fluid Mechanics*, 23:601–639, 1991.
- [4] Y. Levy, D. Degani, and A. Seginer. Graphical Visualization of Vortical Flows by Means of Helicity. *AIAA J.*, 28(8):1347–1352, August 1990.

- [5] C. H. Berdahl and D. S. Thompson. Eduction of Swirling Structure Using the Velocity Gradient Tensor. *AIAA J.*, 31(1):97–103, January 1993.
- [6] J. Jeong and F. Hussain. On the Identification of a Vortex. *J. Fluid Mechanics*, 285:69–94, 1995.
- [7] D. C. Banks and B. A. Singer. A Predictor-Corrector Technique for Visualizing Unsteady Flow. *IEEE Trans. on Visualization and Computer Graphics*, 1(2):151–163, 1995.
- [8] D. Sujudi and R. Haimes. Identification of Swirling Flow in 3D Vector Fields. In *AIAA 12th Computational Fluid Dynamics Conference, Paper 95-1715*, June 1995.
- [9] M. Roth and R. Peikert. A Higher-Order Method for Finding Vortex Core Lines. In *IEEE Visualization '98*, pages 143–150, October 1998.
- [10] R. C. Strawn, D. N. Kenwright, and J. Ahmad. Computer Visualization of Vortex Wake Systems. *AIAA J.*, 37(4):511–512, April 1999.
- [11] I. A. Sadarjoen, F. H. Post, B. Ma, D. C. Banks, and H.-G. Pagendarm. Selective Visualization of Vortices in Hydrodynamic Flows. In *IEEE Visualization '98*, pages 419–422, October 1998.
- [12] M. Jiang, R. Machiraju, and D. S. Thompson. A Novel Approach to Vortex Core Region Detection. In *Joint Eurographics-IEEE TCVG Symposium on Visualization*, pages 217–225, May 2002.
- [13] M. Roth. *Automatic Extraction of Vortex Core Lines and Other Line-Type Features for Scientific Visualization*. PhD thesis, Swiss Federal Institute of Technology Zürich, 2000.
- [14] D. C. Banks and B. A. Singer. Vortex Tubes in Turbulent Flows: Identification, Representation and Reconstruction. In *IEEE Visualization '94*, pages 132–139, October 1994.
- [15] D. N. Kenwright and R. Haimes. Vortex Identification—Applications in Aerodynamics. In *IEEE Visualization '97*, pages 413–416, October 1997.
- [16] D. N. Kenwright and R. Haimes. Automatic Vortex Core Detection. *IEEE Computer Graphics and Applications*, 18(4):70–74, July-August 1998.

- [17] R. Haimes and D. N. Kenwright. On the Velocity Gradient Tensor and Fluid Feature Extraction. In *AIAA 14th Computational Fluid Dynamics Conference, Paper 99-3288*, June 1999.
- [18] R. Peikert and M. Roth. The “Parallel Vectors” Operator—A Vector Field Visualization Primitive. In *IEEE Visualization '99*, pages 263–270, October 1999.
- [19] I. A. Sadarjoen and F. H. Post. Geometric Methods for Vortex Extraction. In *Joint Eurographics–IEEE TCVG Symposium on Visualization*, pages 53–62, May 1999.
- [20] I. A. Sadarjoen. *Extraction and Visualization of Geometries in Fluid Flow Fields*. PhD thesis, Delft University of Technology, 1999.
- [21] H.-G. Pagendarm, B. Henne, and M. Rütten. Detecting Vortical Phenomena in Vector Data by Medium-Scale Correlation. In *IEEE Visualization '99*, pages 409–412, October 1999.
- [22] M. S. Chong, A. E. Perry, and B. J. Cantwell. A General Classification of Three-Dimensional Flow Fields. *Phys. Fluids, A*, 2(5):765–777, 1990.
- [23] A. E. Perry and M. S. Chong. A Description of Eddying Motions and Flow Patterns Using Critical Point Concepts. *Ann. Rev. Fluid Mechanics*, 19:125–155, 1987.
- [24] M. Jiang, R. Machiraju, and D. S. Thompson. Geometric Verification of Swirling Features in Flow Fields. In *IEEE Visualization '02*, pages 307–314, October 2002.
- [25] D. S. Thompson, R. Machiraju, M. Jiang, J. Nair, G. Craciun, and S. Venkata. Physics-Based Feature Mining for Large Data Exploration. *IEEE Computing in Science & Engineering*, 4(4):22–30, July 2002.
- [26] V. Verma, D. Kao, and A. Pang. PLIC: Bridging the Gap Between Streamlines and LIC. In *IEEE Visualization '99*, pages 341–351, October 1999.