An Uncertainty-Driven Approach to Vortex Analysis Using Oracle Consensus and Spatial Proximity

Ayan Biswas*
The Ohio State University
David Thompson†
Mississippi State University
Wenbin He‡
The Ohio State University
Qi Deng§
University of Florida
Chun-Ming Chen¶
The Ohio State University
Han-Wei Shen‖
The Ohio State University
Raghu Machiraju**
The Ohio State University
Anand Rangarajan††
University of Florida

ABSTRACT

Although vortex analysis and detection have been extensively investigated in the past, none of the existing techniques are able to provide fully robust and reliable identification results. Local vortex detection methods are popular as they are efficient and easy to implement, and produce binary outputs based on a user-specified, hard threshold. However, vortices are global features, which present challenges for local detectors. On the other hand, global detectors are computationally intensive and require considerable user input. In this work, we propose a consensus-based uncertainty model and introduce spatial proximity to enhance vortex detection results obtained using point-based methods. We use four existing local vortex detectors and convert their outputs into fuzzy possibility values using a sigmoid-based soft-thresholding approach. We apply a majority voting scheme that enables us to identify candidate vortex regions with a higher degree of confidence. Then, we introduce spatial proximity- based analysis to discern the final vortical regions. Thus, by using spatial proximity coupled with fuzzy inputs, we propose a novel uncertainty analysis approach for vortex detection. We use expert's input to better estimate the system parameters and results from two real-world data sets demonstrate the efficacy of our method.

1 INTRODUCTION

Detection and visualization of flow features in data generated from computational simulations have been important research topics. Along with other flow features, vortices, or eddies, have attracted a great deal of attention from visualization and application researchers. In the past few decades, researchers have analyzed vortex-like structures in the flow and have developed numerous techniques each with its own degree of accuracy. A number of these detectors are point-based and are called local detectors. These detectors classify points in the field by assigning membership to either the class vortex or the class non-vortex based on a local point-wise criterion. Typically, they use a hard-threshold for the classification purposes and the determination of an optimal threshold is necessary for robust detection [6, 11, 37]. On the other hand, non-local vortex detection schemes analyze a region and classify it based on the absence or presence of a vortex. The use of streamlines is very prevalent in this class of detection algorithms. Generally, these algorithms require significant user intervention to perform well. A good survey of vortex detection techniques can be found in [20]. Despite these efforts, there does not exist a robust and reliable method that detects vortex structures. Further, the accuracy of different detectors varies from data set to data set and, with the continued growth of the size and complexity of simulation data, the need for a better automated vortex analysis and detection scheme is paramount. We propose an approach that leverages the strengths of individual local detectors while exploiting global information to optimize the performance of the chosen detectors.

Modeling uncertainty has gained popularity in the past decade and has found applications in flow visualization. For vortex detection, analysis was previously conducted using probabilistic methods assuming the data contained uncertainty [24]. However, in our work we propose to analyze the uncertainty that arises from the differences in the performance of existing vortex detection schemes, which act as inconsistent oracles. That is, there exists a mismatch among the regions identified as vortices by different detectors. Moreover, the threshold values used by these different detectors provide a reference related to the degree of confidence of the prediction. If the output from a detector is very close to its theoretical threshold, then the prediction contains more uncertainty. This observation enables us to model this uncertainty as a fuzzy membership problem for the vortex and non-vortex classes in the flow field. A second characteristic that we exploit is *locality*. A vortex is defined as a contiguous region within the domain. A study of vortical features reveals that a point has a higher chance of being part of a vortex if it is located near pre-identified regions of a vortex while a point that is not located near a vortex is much less likely to be part of a vortex. In essence, we seek confidence from the co-occurrence of similar events and classified points and further improve the performance of local detectors with more global information.

In this work, we introduce a novel vortex analysis workflow that enables the users to analyze and extract vortices with more accuracy and robustness. We use four existing local vortex detection techniques as the basis of identification process. For each of the detectors, we apply a logistic function that converts the output of each detector to a fuzzy *possibility value* in the range of 0 to 1. Using the method of entropy maximization, we allow for parameters to be automatically selected for this conversion. The possibility values generated at this stage are used for a voting scheme to identify the regions of a candidate vortex structure with higher degree of certainty. Utilizing the property of spatial locality, uncertain points are then classified into either vortex and non-vortex classes. In our workflow, a domain expert provides her/his input by marking a small number of sample points, which are used for estimating the parameters of the system based on global information. To the best of our knowledge, our approach is the first attempt to incorporate spatial locality with uncertainty analysis for the detection of vortical features.

Our contributions are threefold:

1. We introduce a new approach towards modeling the uncertainty of the existing vortex detectors by using their threshold values to create a fuzzy input-based system.

*e-mail: biswas.36@osu.edu
†e-mail:dst@cavs.msstate.edu
‡e-mail: he.495@buckeyemail.osu.edu
§e-mail: contact.qdeng@gmail.com
¶e-mail: chenchu@cse.ohio-state.edu
‖e-mail: hwshen@cse.ohio-state.edu
**e-mail: raghu@cse.ohio-state.edu
††e-mail: anand@cise.ufl.edu

2. We present a consensus based voting method to identify vortex regions with a higher degree of confidence and segregate the different certainty levels for different spatial regions.

3. We introduce the use of spatial proximity into the vortex detection framework. Different vortex clusters are identified and a distance-based criterion is used to assign the final vortex points in the system.

2 Related Work

Vortex detection algorithms are generally classified as one of two types, either local or global, depending on the neighborhood of the data upon which the classification is based [33]. Many of the local vortex detectors are based on the Jacobian of the velocity field and classify individual points of the domain as vortex based on a point-wise criterion. The Q-criterion [17], the λ_2-method [18], the Δ-criterion [7], and the Γ_2 method [14] are examples of local detectors that only look the Jacobian of the velocity field to decide vortical flows. Although Schafhitzel *et al*. [31] argued that the λ_2 method is more reliable and effective in comparison to other methods, its performance varies from data set to data set [36] and all of the methods produce false positives and false negatives. Roth and Peikert [29] presented the parallel vectors operator to locate vortex core lines. To capture the slowly rotating vortices, they used the second derivative of the velocity field. Other approaches have also been proposed for the extraction based on parallel vectors [26, 28].

Global methods generally use streamlines/pathlines to detect vortical regions. Sadarjoen *et al*. [30] used the winding angle method to detect vortices using streamlines. Jiang *et al*. [19] proposed another streamline-based method whereby they analyzed the geometry of the streamlines to identify the vortex core regions. Banks and Singer [2] proposed a predictor-corrector method for the extraction of vortex core lines. Haller [15] proposed an objective definition for a vortex that is applicable in a rotating reference frame. It is similar to the Q-criterion except that it incorporates global information via a Lagrangian analysis. Pagendarm *et al*. [25] calculated certain local geometric properties of streamlines and then employed a model vortex to estimate the curvature density field. The curvature density field was then visualized using isosurfaces to identify vortex core regions. In a recent work, Kohler *et al*. [22] proposed a semi-automatic method for vortex detection using line predicates and the λ_2-method to visualize blood flow.

Analysis and quantification of uncertainty in flow visualization have received much interest in the last decade. Otto and Theisel proposed a method [24] to analyze vortices in the scenario where the underlying data set contains uncertainty. Assuming that the uncertainty follows a Gaussian distribution, they used Gaussian fitting in the neighborhood of a point and used a Monte-Carlo sampling approach to approximate the vortex detector methods at a given point. Burger *et al*. [5] previously used fuzzy sets for more intuitive visual exploration of different vortex detection methods but they did not provide a framework that combined the vortex detectors. In our work, instead of assuming the uncertainty is manifest in the data set, we propose to explore the uncertainty that arises from the use of multiple vortex detectors and their output values.

Machine learning-based methods [16] are gaining popularity for solving many graphics and visualization related problems [13, 21, 23, 37]. Recently Zhang *et al*. [37] employed the Adaptive Boosting [12] approach to assign weights to the different vortex detection schemes and showed improvement in the error rates at the expense of more false positives. In our work, instead of assigning weights to the individual detectors, we propose to assign equal weights and introduce a distance-based criterion to further improve the results.

3 Vortex Detection Techniques

3.1 Global Streamline Method

Global methods for vortex detection generally attempt to find a coherent region in the field with closed or spiraling streamlines in a reference frame moving with the vortex. An intuitive, streamline-based definition of a vortex was provided by Robinson [27] as: "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." This definition encapsulates the inherently global nature of a vortex. Unfortunately, it is self-referential and difficult to describe algorithmically. Determining the appropriate reference frame moving with the vortex is a non-trivial issue [35]. Additionally, seeding the streamlines at the correct location is of utmost importance. Analysis of the geometric properties of a streamline [19, 25] or computation of the winding angle [30] are two examples of streamline-based vortex detection methods.

3.2 Local Vortex Detectors

Local vortex detectors evaluate a function at each point of the field and generate a numerical value. This local field value is used to decide whether the point is a member of a vortex region with the aid of a threshold function. Different local detectors have disparate threshold functions and the choice of the threshold plays an important role [6, 37, 11] in determining the efficacy of a given vortex detection algorithm. Most local vortex detectors are based on the velocity gradient tensor J, where $J = \bigtriangledown v$. For this reason, these detectors are Galilean invariant or invariant under translation [20]. The estimated velocity gradient tensor is used to compute the rate-of-strain tensor $S = \frac{1}{2}(J + J^T)$ and the rate-of-rotation tensor $R = \frac{1}{2}(J - J^T)$.The four most popular local detectors include the Q-criterion, λ_2, Δ-criterion, and Γ_2. After generating values at each point, a function L can be defined for each of these detectors with $L : \mathbb{R} \longrightarrow \mathbb{B}$, where $\mathbb{B} \in \{0, 1\}$. The function L is used for labeling the flow field points with 1 as a vortex label and 0 as a non-vortex label. This function depends on the threshold of the detectors and their operating characteristics.

The Q-criterion was proposed by Hunt *et al*. [17]. It identifies a vortex by finding the regions where the vorticity of the field exceeds the rate of strain, i.e., rotation dominates deformation. In addition, the pressure in this region needs to be lower than the ambient pressure implying the presence of a rotation-induced pressure minimum. Thus, the Q-criterion is computed as $Q = \frac{1}{2}(\| R \|^2 - \| S \|^2)$. The corresponding labeling function L_Q is defined as:

$$L_Q(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{if } x \leq 0 \end{cases}. \tag{1}$$

Jeong and Hussain [18] proposed the λ_2 method for vortex detection by extracting a connected region wherein the matrix $S^2 + R^2$ has two negative eigenvalues. If the second largest eigenvalue of this matrix at a given point in the field is negative, then this point is located in a region with a rotation-induced pressure minimum and belongs to a vortex core region. Following this method, the labeling function is defined as:

$$L_{\lambda_2}(x) = \begin{cases} 1, & \text{if } x < 0 \\ 0, & \text{if } x \geq 0 \end{cases}. \tag{2}$$

Chong *et al*. [7] defined the Δ-criterion for vortex detection using critical point theory. In this definition, a vortex core is identified as a region where the Jacobian J has complex eigenvalues, which would produce periodic trajectories in the region near the critical point. The Δ-criterion is defined as $\Delta = (\frac{1}{2}P)^2 + (\frac{1}{3}Q)^3$ where $P =$

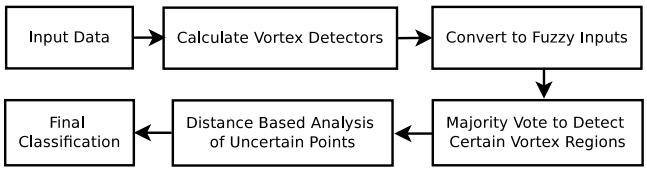


Figure 1: A schematic view of the different stages of our system.

$Det(J)$ and $Q = \frac{S_{ij}S_{ji}+R_{ij}R_{ji}}{2}$. The labeling function for this method is defined as:

$$L_{\Delta}(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{if } x \leq 0 \end{cases}. \quad (3)$$

Graftieaux *et al.* [14] proposed the Γ_2 method for detecting the vortex boundary. This method identifies the regions where rotation dominates strain by looking at the maximum eigenvalues of the rotation tensor and strain rate tensor. This method can be implemented by considering the ratio $\| r \| / \| \mu \|$ where $\| r \|$ and $\| \mu \|$ denote the maximum eigenvalues of the rotation tensor matrix R and the strain rate tensor S respectively. The labeling function is given as follows:

$$L_{\Gamma_2}(x) = \begin{cases} 1, & \text{if } x > 1 \\ 0, & \text{if } x \leq 1 \end{cases}. \quad (4)$$

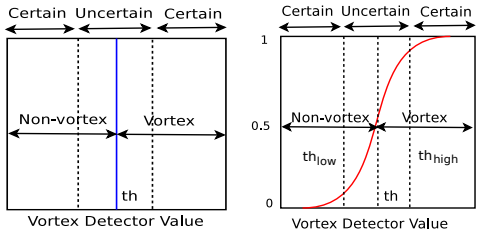
4 System Overview

It is our assertion that Robinson's definition, i.e., the notion of a swirling flow, is the most fundamental description of a vortex. In fact, as previously noted by Pagendarm [25], most vortex detection methods employ proxy variables that are easily defined from a mathematical perspective and have some level of correlation, albeit not perfect [31, 36], with the occurrence of a vortex. It is not our intent to debate the relative merits of the different techniques. Our goal is to improve the robustness of the vortex detection process by combining several existing, computationally-efficient local vortex detectors into a more robust detector that incorporates global information through a distance function that is based on the expert labels. To achieve this, we initially compute a set of vortex detectors from the input data set. In our interpretation, the signs and values of each of the different existing vortex detectors represent different degree of *vortexness* for a point. We convert their outputs to a fuzzy value from 0 (certain non-vortex) to 1 (certain vortex) using a sigmoid function. This fuzzy conversion allows us to combine these vortex detectors by applying a majority vote which, in turn, identifies the regions of the data set that are more likely to be part of a vortex. In the next stage of the pipeline, the remaining uncertain points are classified according to their proximity from the vortex regions, which are detected from majority voting, and a distance-based fall-off scheme is applied to adjust their vortexness values for final classification. In this generic framework, there are two parameters that need to be tuned. We work closely with a domain expert who provides her/his inputs that are used for estimating these parameters. This enables us to incorporate global domain knowledge into our system with increased robustness and accuracy. Figure 1 represents a schematic view of the different stages of the pipeline.

5 Algorithm

5.1 Fuzzy Initiation of the Vortex Detectors

In the vortex detection problem, if a pointwise detector's output is close to its theoretical threshold value, then the classification is generally more uncertain compared to the case where the value is further away from the threshold. This is represented schematically in Figure 2a with th as the threshold value. The hard-thresholding



(a) Certain and Uncertain regions after hard-thresholding around th.

(b) Uncertainty in the output is modeled using a sigmoid curve.

Figure 2: Uncertainty in vortex detection and modeling via sigmoid curve.

of the pointwise detectors ignores this property and generates binary 0 or 1 as an output where 0 denotes non-vortex regions and 1 denotes the presence of a vortex. To incorporate the uncertainty that is inherent to these vortex detectors, instead of generating hard-thresholded 0 or 1 output from the existing detectors, it is more desired to generate fuzzy outputs in the interval $[0, 1]$ such that it represents the *vortexness* of the point given the detector.

Previously Burger *et al.* [5] chose a linear mapping function to compute the vortexness. In our work, we use the sigmoid function F which is given as:

$$F(x) = \frac{1}{1 + e^{-a(x-th)}} \quad (5)$$

where the parameter th denotes the point of maximum uncertainty: when $x = th$, $F(x) = 0.5$. The parameter a represents the fall-off rate or the steepness of the sigmoid curve. We use a sigmoid function for the following reasons: 1) The sigmoid function is more general approach compared to linear mapping. Further, the sigmoid function can be used to imitate the linear mapping through proper choice of the parameters. 2) Classification problems are inherently sigmoidal. To illustrate this point, let us consider two classes C_1 (vortex class) and C_2 (non-vortex class) and an observation x (output of a vortex detector), the posterior probability of classification $p(C_1|x)$ can be written [4] following the Bayesian approach as:

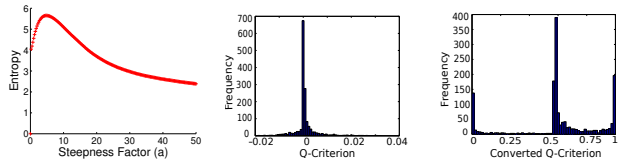
$$p(C_1|x) = \frac{p(x|C_1)p(C_1)}{p(x|C_1)p(C_1) + p(x|C_2)p(C_2)} = \frac{1}{1 + e^{-a}} \quad (6)$$

where the value of a is defined as $a = \ln \frac{p(x|C_1)p(C_1)}{p(x|C_2)p(C_2)}$. In Equation 6, $p(C_1|x)$ follows a sigmoid or S-shaped distribution with a steepness factor a as shown in Figure 2b.

For vortex detection, the sigmoid function F can be applied to the outputs of the individual vortex detectors to convert them to fuzzy *vortexness* values. After the fuzzy conversion, values closer to 1 indicate a higher possibility to be part of a vortex. If the output value is close to 0, then it is more likely to be marked as non-vortex. The values around 0.5 are the uncertain values. Figure 2b shows a schematic view of this conversion. Given a collection of vortex detectors, we assign one fuzzy conversion function F to each individual detectors by tuning the parameters th and a as discussed in the next section, Section 5.2.

5.2 Parameter Selection Using Information Theory

The fuzzy conversion function $F(x)$ consists of two parameters, th and a, as shown in Equation 5. Since each individual vortex detector is assigned a function $F(x)$ to convert to the fuzzy output, the parameters of $F(x)$ also depend on the properties of the individual detector. Since the output of $F(x)$ is most uncertain when $x = th$, the value of th is set to the theoretical threshold value of that vortex detector which is known for each detector as described earlier.



(a) Change of entropy with the change in steepness parameter a.

(b) Original distribution of Q-criterion before applying entropy maximization.

(c) Distribution of Q-criterion after applying entropy maximization.

Figure 3: Change in the distribution due to entropy maximization.

This describes the property that the pointwise detectors are most uncertain around their theoretical threshold.

The steepness parameter a has two components: the sign and the magnitude. Since the negative values for some of the detectors are more likely to be a vortex (e.g., λ_2) whereas for others, the positive values (e.g., Q-criterion) represent vortex regions, the sign of a is chosen to make all the detectors consistent after fuzzy conversion. The sign is determined by using the function S as follows:

$$S(a) = \begin{cases} +, & \text{if } L(x) = 1,\ \forall x > th \\ -, & \text{if } L(x) = 0,\ \forall x > th \end{cases} \tag{7}$$

where for a given vortex detector, L is the previously defined labeling function, th is corresponding theoretical threshold and $x \in \mathbb{R}$.

After determining the sign, the magnitude of a must be determined. If $|a| \approx 0$, then after the fuzzy conversion of the vortex detector, the histogram of the resulting vortexness values shows a very high peak near 0.5. This reflects that, after fuzzy conversion, all the values of the vortex detector have mapped to the uncertain 0.5 in the fuzzy scale. Similarly, if $|a|$ is very large, then the resulting histogram of vortexness values reduces to two peaks, at 0 (certain non-vortex) and 1 (certain vortex), representing the hard-thresholding scheme. We strive for a value of the steepness factor a that generates a vortexness value distribution that is between these two extremes such that the converted fuzzy vortexness values are well separable. This conversion will represent the confidence of the vortex detector and to achieve a well-distributed histogram after fuzzy conversion, we apply the notion of entropy maximization [3, 8]. For a random variable X, if $p(x)$ denotes the probability of occurrence of an observation $x \in X$, then entropy $H(X)$ is defined as

$$H(X) = -\sum_{x \in X} p(x) \log p(x). \tag{8}$$

An increase in the entropy value for a random variable generally signifies that the values of the variable are well distributed in the different bins of its histogram. In our entropy maximization method, the parameter a is chosen such that it maximizes the entropy of the distribution of resulting fuzzy vortexness values of the detector after the application of the sigmoid function. Figure 3 shows results from rearward facing step data set [1], which is described in more detail in Section 6.2. In this example, the application of entropy maximization changes the unimodal distribution of detector values into a distribution with three peaks. Instead of simply applying entropy maximization, we use the parameter value that maximizes the product of entropy and the rate of change of entropy to reduce the points mapped to the values 0 and 1. This enhances the separation of data points based on the transformed values in the following stages and thus increasing the efficiency of the algorithm.

5.3 Learning From Multiple Oracles

Since the individual vortex detectors can be inconsistent in their identification of a vortex, they can be treated as the imperfect oracles of our vortex detection system. It has been shown [32, 34, 37] that the use of multiple detectors instead of a single detector, provides a higher level of reliability for vortex detection and visualization. After converting the outputs of the individual vortex detectors to represent their fuzzy confidence levels, we propose to aggregate their prediction in a manner similar to the fuzzy "AND" operation to quantify the agreement among themselves. To implement this fuzzy "AND" operation given four vortex detectors, we apply a majority vote algorithm as :

$$f(x_1,x_2,x_3,x_4) = \begin{cases} 1, & \text{if } \sum_{i=1}^{4} Greater(x_i, th_{high}) \geq 3 \\ 0, & \text{if } \sum_{i=1}^{4} Greater(th_{low}, x_i) \geq 3 \\ -1, & \text{Otherwise} \end{cases}. \tag{9}$$

Here, th_{high} and th_{low} are two threshold values such that $0.0 < th_{low} < th_{high} < 1.0$ and $Greater$ is a boolean comparison operator which is defined as:

$$Greater(a,b) = \begin{cases} 1, & \text{if } a \geq b \\ 0, & \text{if } a < b \end{cases}. \tag{10}$$

Here th_{high} denotes the threshold above which the points can be marked as vortex with a high degree of certainty. Similarly, th_{low} denotes the threshold value below which the points can be marked as non-vortex with higher confidence. The values falling between these two thresholds are considered uncertain points that need further attention. Thus, Equation 9 defines a majority voting scheme based on these two parameters. For a given point, if at least three of the four detectors assign a vortexness value greater than th_{high}, then this point is marked with 1. Similarly, if three out of four detectors provide a value less than th_{low}, then this point is initially marked with 0. The rest of the uncertain points are assigned the label -1 and these points are then further processed according to their proximity from the already detected certain vortex points as discussed in Section 5.4. The estimation of the thresholds th_{high} and th_{low} is discussed in Section 5.5.

5.4 Knowledge Enhancement Through Spatial Proximity

Analysis of individual data points alone, to determine their likelihood of being part of a vortex, limits the effectiveness of the detection process. We propose that the incorporation of spatial locality into the pointwise vortex detectors may enhance the capabilities of the detector by making it less localized. If it is certain that one point is part of a vortical flow, then the likelihood of its neighbors being in a vortical flow also increases. On the contrary, if it is known that a point is surely a non-vortex, then its neighbors also become more likely to be non-vortex. This spatial locality feature can be incorporated into our system as follows. After identifying the more certain candidate vortex regions with the oracles as shown in Section 5.3, we first perform spatial clustering of these points. Thus, we have in effect identified salient points near the vortex core. The clustering is performed using a "region fill" approach in which each point labeled as a vortex starts as an individual cluster, which is then merged with the neighboring clusters and, in turn, grown into even larger clusters. Next, we analyze the remaining uncertain points based on these clusters. After the identification of the clusters, we take advantage of the spatial locality for the uncertain points that have the label -1. For each of these points, we determine the nearest vortex cluster and calculate the distance d from that cluster. The vortexness values of these uncertain points are readjusted according to a distance based decay function, D.

$$D(d) \propto \frac{1}{d^2} \Leftrightarrow D(d) = \frac{K}{d^2}. \tag{11}$$

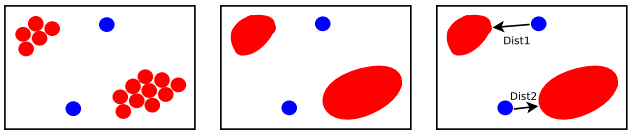


Figure 4: Incorporation of spatial proximity. (a) Certain (red) and uncertain (blue) points detected after majority voting. (b) Spatial clustering of certain points. (c) Uncertain points are classified according to their distance from their nearest certain vortex cluster.

Decay is modeled as an inverse square function. The proportionality constant K is dependent on the data and is estimated using domain expert's input for higher accuracy as discussed in Section 5.5. Suppose, x_{avg} denotes the average vortexness value of an uncertain point calculated from the four vortex detectors. Now, for all the uncertain points, x_{avg} is adjusted based on distance and these uncertain points are finally classified according to the previous high-threshold th_{high}.

$$f(d, x_{avg}) = \begin{cases} 1, & \text{if } x_{avg} * (1 + \frac{K}{d^2}) \geq th_{high} \\ 0, & \text{Otherwise} \end{cases}. \quad (12)$$

Figure 4 illustrates this stage of the framework. The individual certain vortex points (marked in red) of Figure 4a are spatially clustered in Figure 4b. The uncertain points (marked in blue) are then recalculated for their vortexness according to their distance from the nearest cluster as shown in Figure 4c.

5.5 System Integration with Domain Expert's Input

In this section, we integrate the different components of our system and elaborate the role of the domain expert. We worked closely with a domain expert (also a co-author for this work) who employed a global streamline method to generate the labels for points in the field which will be used to quantitatively measure and compare our proposed method with the existing ones. Our approach is similar to the one described in [37] that is based on Robinson's definition of a vortex [27] and the observation that a Galilean transformation to an appropriate reference frame is necessary for unsteady flows with moving vortices [35]. The domain expert uses a manual vortex detection process whereby a set of streamlines is generated from seed points placed by the expert. Since streamlines are not Galilean invariant, the domain expert must adjust the local reference frame [35] in an iterative process to make sure that the swirling nature of the streamlines is accurately captured. The goal of the expert is to select a set of parameters that produce a coherent set of streamlines. To aid in this process, the domain expert initially uses a point based detector, such as the λ_2 method to generate isosurfaces that identify the candidate vortical regions in the field. The domain expert then employs a "point picking interface" by which he can select regularly-shaped regions in the field to explore further with streamlines. If the domain expert observes a coherent set of swirling streamlines, then he changes the view point such that the vortex axis is aligned with the view plane normal. Then these spatial three-dimensional points are marked with 1 for vortex and 0 for non-vortex. Further refinement can be obtained by picking individual points and changing their labels.

With the expert labels available, we now integrate all the different components of our system in this section. For a given data set, the four vortex detectors- Q-criterion, λ_2 method, Δ criterion, and Γ_2- are first calculated. These outputs are converted to fuzzy values by using the sigmoid functions that maximize the product of entropy and rate of change of entropy of their histograms as described in Sections 5.1 and 5.2. Then majority voting is applied based on the two thresholds: th_{high} and th_{low}. The low threshold th_{low} is set to 0.25 for all the experiments. For increased accuracy, the high threshold th_{high} is estimated from a small fraction of the domain expert's labels. At this stage, a small number of samples (typically 10%) are randomly chosen and th_{high} is selected such that the false positives are minimized given these random samples. The data points are now marked with certain 0 or 1 labels and uncertain -1 labels. In the next stage of our pipeline, the certain 1 values are spatially clustered. Then all of the uncertain -1 data points are separated and their distances from the nearest vortex cluster are computed. Based on these nearest distances, the uncertain points are classified. The proportionality constant K in Equation 11 is also estimated from the same set of samples drawn from the domain expert's labels such that the final error rate and the true positive rates are higher than each individual detector's performance. Finally, the remainder of the domain expert's labels are used as a basis of comparison for generating results to quantitatively assess the effectiveness of the different vortex detection methods.

6 Results

The experiments were conducted on a Linux machine with 2.40 GHz Intel core i7 CPU, 8 GB of RAM and an NVIDIA Geforce GT 650M GPU with 2GB texture memory. For generating comparative results, we use the labels generated by the domain expert as a basis of comparison for the below mentioned data sets. The following metrics are used for quantitative comparison among different vortex detection techniques:

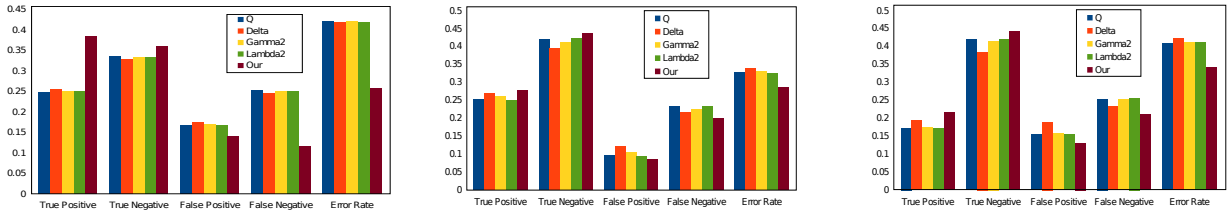
$$\begin{aligned} ErrorRate &= (P_F + N_F)/(T+N) \\ TruePositiveRate &= P_T/(T+N) \\ TrueNegativeRate &= N_T/(T+N) \\ FalsePositiveRate &= P_F/(T+N) \\ FalseNegativeRate &= N_F/(T+N) \end{aligned} \quad (13)$$

where P_T denotes the true positive count, N_T denotes the true negative count, P_F is the false positive count, N_F is the false negative count, T is the total vortex count in expert labels and N is the total non-vortex points in expert labels. The parameters of our system, high threshold th_{high} and distance based constant K, are tuned using random samples (10%) from the expert labels with the performance of the algorithms measured on the rest of the labeled points.

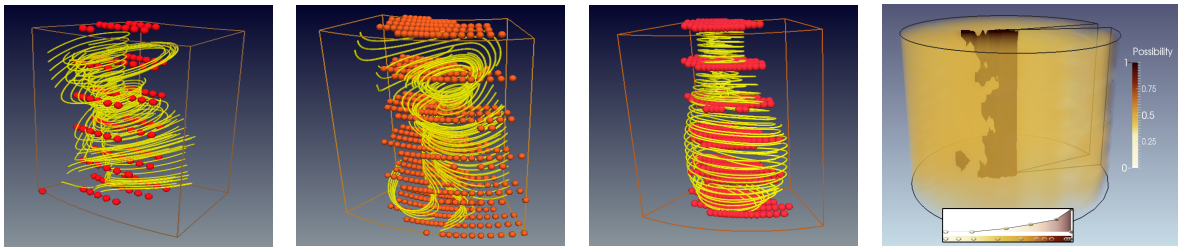
6.1 Tapered Cylinder

Our method is first applied to the Tapered Cylinder data set [9], which describes an unsteady, three-dimensional, incompressible, laminar, viscous flow around a cylinder with its axis oriented perpendicular to the primary flow. It should be noted that, since the cylinder extends across the entire domain, there are no tip effects present in the simulation.

We compare our proposed method with the four existing vortex detection algorithms used as oracles for our system : Q-criterion, λ_2 method, Δ criterion, and Γ_2 method. Figure 5a shows comparative results for the time step 12200 of this data set with Y-axis showing the percentage count of the data points. It is readily visible that, for this time step, our proposed method significantly increases the true positive rate and reduces the total error rate. It is also observed that the true negative rate is better than any of the four existing detectors and false positive and false negative rates are consequently reduced. Figure 5d represents an example where the domain expert used the global streamline method to generate the labels for this time step. Here, the red spheres indicate the expert's vortex labels and streamlines were seeded in the region after adjusting the reference frame. For more details about the streamlines and selection of reference frame, we refer the interested readers to the supplemental material. Our improved error rate confirms that our method produces more consistent with the domain expert's labelings. Experiments were further conducted on other time steps, which are far apart in the

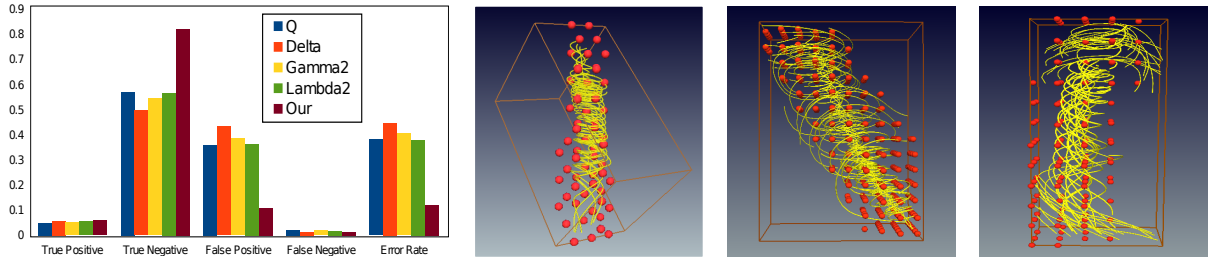


(a) Time step 12200. (b) Time step 12780. (c) Time step 15490.



(d) Expert labels for 12200 (e) Expert labels for 12780 (f) Expert labels for 15490 (g) Visualization of our prediction field

Figure 5: Results for different time steps of the Tapered Cylinder data set.



(a) (b) (c) (d)

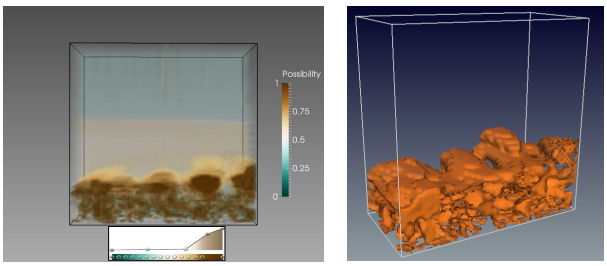
Figure 6: Results for the Rearward Facing Step data set. (a) Quantitative comparison results. (b)-(d) Domain expert's labeled points in three regions of the data set.

temporal domain. Figure 5b and Figure 5c show the results obtained from 12780 and 15490 respectively. In each of these time steps, our proposed method provides improved results for all types of errors. Examples of the domain expert's labelings are shown in Figure 5e and 5f for time steps 12780 and 15490 respectively. The results show a good agreement with the domain expert's markings which in turn shows more robust result generated by our method. The false positive rates of the points that were decided by spatial proximity criteria are 0.09, 0.05 and 0.11 respectively for the time steps 12200, 12780 and 15490. In Figure 5g, the volume rendering of the final prediction field generated from our algorithm for the time step 12200 is shown. Here the vortex regions of higher confidence are suitably highlighted by setting the transfer function.

6.2 Rearward Facing Step

We now apply our method to a data set that models the unsteady, incompressible, turbulent flow over a rearward facing step. The flow enters the domain from the lower left in the positive x direction, encounters the step, and is separated. The region of separated flow is highly turbulent and contains vortices. In this case, the axis of a given vortex may have significant curvature and its orientation with respect to the primary flow may vary widely from other nearby vortices. The flow conditions were chosen to match the experimental data obtained by Driver and Seegmiller[10]. Details concerning the numerical simulation are reported in Alam *et al.* [1].

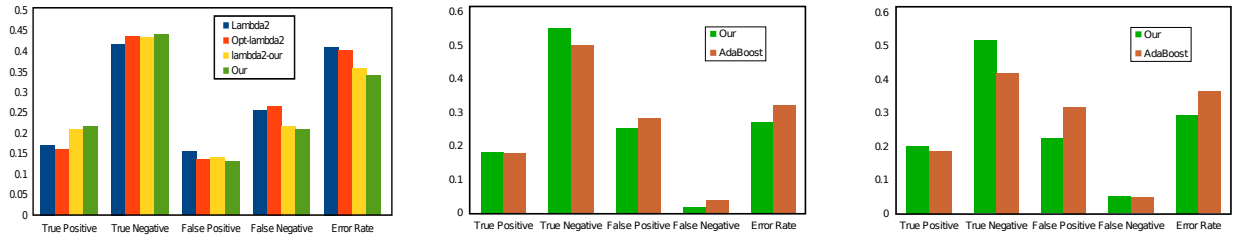
Figure 6a shows a comparison of the four vortex detectors and our proposed method to provide a complete quantitative analysis. As can be seen from this plot, the overall true positive rate is slightly



(a) (b)

Figure 7: Results for the Rearward Facing Step data set. (a) Volume rendering from our algorithm generated prediction field. (b) Isosurface from the prediction field showing the vortical regions detected from our algorithm.

better than the existing detectors, since the total number of vortex points is smaller in this data set, but the major performance gain is achieved in the true negative rate. Our proposed method was able to eliminate most of the false positive labels and thereby significantly reduce the total error rate. Figure 6b, Figure 6c and Figure 6d show three regions in the data set where the domain expert extracted the voritcal motion by adjusting the correct reference frame (more details provided in the supplemental material). Here, the red spheres indicate the expert's vortex labels. Figure 6a shows that our results conform to the domain expert's labels and was able to identify the vortices with improved accuracy. For this data set, the false positive rate of the points that were classified by the spatial proximity criteria was 0.12, which is close to the overall false positive rate



(a) (b) (c)

Figure 8: Comparison results for the Tapered Cylinder data set. (a) Comparison with different modifications of our proposed algorithm from time step 15490. (b) Comparison with AdaBoost from time step 12210. (c) Comparison with AdaBoost from time step 12240.

as shown in Figure 6a. Figure 7a shows the volume rendering of the resulting possibility field of a region of this data set with the transfer function shown below that emphasizes the vortex regions of higher confidence. We note that the bottom part of the data set is more turbulent. Finally we show the isosurface generated from our prediction field in Figure 7b depicting the detected vortex regions.

7 Comparison and Discussion

In this section, we compare the results of our proposed framework with other techniques that benefit from the labels generated by the domain expert. In Figure 8a, the four different scenarios are presented from time step 15490 of the Tapered Cylinder data set. In this figure, one set of results is taken after generating the optimized hard-threshold [37] of an existing detector, namely the λ_2 method as it is widely used given its reliability compared to the other pointwise metrics. "Opt-Lambda2" depicts this optimized threshold selection in the figure with "Lambda2" as the theoretical hard-thresholded λ_2 example. Next, we modify our proposed algorithm and, instead of using a consensus of four existing detectors, we use λ_2 as the only input to our system, keeping all the other stages of our proposed method the same. This situation is shown as "lambda2-our" in Figure 8a. For comparison purposes, our complete proposed algorithm is also shown in the figure and is labelled as "Our". The results show that the use of an optimized threshold improves the error rate but at the expense of a reduced true positive rate. Both "lambda2-our" and "Our" outperform "Lambda2" and "Opt-lambda2", which suggests that the use of spatial locality information is an important factor for effective detection of vortices. The comparison of "Our" and "lambda2-our" shows that consensus from four oracles is better than trusting just one, especially when all of the oracles are fallible and the best oracle for a given data set is not known a priori. The exclusive use of any of the other three existing metrics instead of λ_2 produces similar results.

Next we compare our method with the AdaBoost method, which was recently proposed by Zhang *et al.* [37]. This AdaBoost method also uses multiple vortex detectors as their input and tries to optimize the performance using samples from domain expert. To conduct the experiment, we selected the previously mentioned 12210 time step from the tapered cylinder data set. In the Figure 8b, "Our" denotes the method proposed in this paper and "AdaBoost" is the method proposed in [37]. To estimate the system parameters for both the methods, 10% of the domain expert's markings from the data were used and the results were generated. Also, to test the reliability and sensitivity of the two methods, we conducted another set of experiments in which we used 10% of the domain expert's markings selected only from time step 12210 to tune the system parameters and applied the estimated parameters to the later time step 12240. Here we assume that the time-varying flow field properties will be changing slowly over the time interval and the parameters estimated from the previous step should be applicable to the next with only a slightly increased error rate. The results for these are indicated by "Our" and "AdaBoost" in Figure 8c. From Figure 8b and 8c, it can be observed that even if the parameters are tuned at each time step or if the parameters are estimated from an initial time step and re-applied to a later time step, our method outperforms the AdaBoost method.

We now discuss the performance of different components in our proposed framework as shown in Table 1. Here, TC refers to Tapered Cylinder data set and RFS denotes the Rearward Facing Step data set. In the first column, we present the time taken to generate the four vortex detection methods from the curvilinear grid data sets. The time taken at this stage depends on the data size and complexity of the gradient computation. The next stage denotes the mutual information maximization stage that is used to convert the four vortex oracles to fuzzy input. We used the *fminsearch* function provided by Matlab at this stage. The next column shows the majority voting scheme which polls the four oracles to decide the certain region. The last column presents the time for distance computation of the uncertain points from the more certain vortex regions. This stage uses the NVIDIA Thrust library to exploit GPUs for efficient distance computation. The computation time of this stage depends on the data size and the number of uncertain points together with the more certain clusters. Using *fminsearch*, this stage requires four to five iterations for our case.

Table 1: Running time for different components of the framework.

	Vortex Data Generation (Sec.)	Mutual Information Maximization (Sec.)	Majority Vote Scheme (Sec.)	Cluster Distance Computation (Sec.)
TC	33.8	4.5	0.007	24.5
RFS	61.3	8.3	0.021	71.7

8 Limitations and Future Work

Our system produces improved results when applied on two different data sets. There still remain some issues and future improvement possibilities. The first concern is the reliance on the domain expert's labels to optimize the system parameters. Currently, for a given data set, we have used labels from a single expert. If there are multiple experts providing their labels, then we will need to formulate a strategy to combine the experts' labels. We keep this as a part of our future work. Also, the process of generating the labels involves manual work from the domain expert and we intend to alleviate this problem by designing more improved systems that can make domain expert's work easier by automatically determining the appropriate reference frame, which is a non-trivial research topic. Finally, we currently process all the time steps individually and we do not use the fact that detecting vortices from later time step can benefit from the results at a previous time step. As a next step, we want to extend our work to track the vortex features over time and analyze how they evolve. Eventually, we want to incorporate our proposed distance-based feature in machine learning methods to enhance the accuracy of the vortex detection process.

9 Conclusion

In this work, we propose a novel vortex analysis framework that examines the uncertainty contained in four existing local vortex detection methods and combines them to perform the vortex analysis and detection in a more robust way. Since there are multiple vortex detectors available to us, we use these detectors as the inputs to our system and model the uncertainty of their predictions using a sigmoid function to convert the outputs of these detectors to a possibility value range of 0 and 1. These possibility values denote the certainty that a point will be classified as a part of a vortex region. We apply a voting algorithm to classify the more certain vortex regions and cluster these regions based on their spatial locations. Next we introduce the use of spatial proximity and classify the remaining points as vortex or non-vortex based on their distance from the closest vortex cluster. We worked closely with a domain expert and used the data marked by the expert to compare our method with other existing methods. We applied our method on multiple data sets and time steps and showed that the accuracy can be improved by systematically analyzing these flow structures.

Acknowledgements

This work was supported in part by NSF grants IIS- 1250752, IIS-1065025, and US Department of Energy grants DE- SC0007444, DE-DC0012495, program manager Lucy Nowell.

References

[1] M. Alam, D. Walters, and D. Thompson. Evaluation of a dynamic hybrid rans/les modeling methodology for attached and separated flows. *ASME Journal for Fluids Engineering (submitted)*.

[2] D. Banks and B. Singer. A predictor-corrector technique for visualizing unsteady flow. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):151–163, June 1995.

[3] R. Benzid, D. Arar, and M. Bentoumi. A fast technique for gray level image thresholding and quantization based on the entropy maximization. In *Systems, Signals and Devices, 2008. IEEE SSD 2008. 5th International Multi-Conference on*, pages 1–4, July 2008.

[4] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[5] R. Bürger, P. Muigg, M. Ilcik, H. Doleisch, and H. Hauser. Integrating local feature detectors in the interactive visual analysis of flow simulation data. In *Proceedings of Eurographics/ IEEE-VGTC Symposium on Visualization 2007*, pages 171–178, 2007.

[6] P. Chakraborty, S. Balachandar, and R. J. Adrian. On the relationships between local vortex identification schemes. *Journal of Fluid Mechanics*, pages 189–214, 7 2005.

[7] M. S. Chong, A. E. Perry, and B. J. Cantwell. A general classification of three-dimensional flow fields. *Physics of Fluids A: Fluid Dynamics*, 2:765–777, 1990.

[8] F. Dambreville. Combining evidences by means of the entropy maximization principle. In *Information Fusion, 2007 10th International Conference on*, pages 1–8, July 2007.

[9] C. L. Dennis C. Jespersen. Numerical Simulation of Flow Past a Tapered Cylinder. In *29th Aerospace Sciences Meeting*, 1991. AIAA Paper 91-0751.

[10] D. Driver and H. Seegmiller. Features of a reattaching turbulent shear layer in divergent channel flow. *AIAA journal*, 23:163–171, 1985.

[11] Y. Dubief and F. Delcayre. On coherent-vortex identification in turbulence. *Journal of Turbulence*, page N11, 2000.

[12] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, pages 119–139, Aug 1997.

[13] R. Fuchs, J. Waser, and M. Groller. Visual human+machine learning. *Visualization and Computer Graphics, IEEE Transactions on*, pages 1327–1334, Nov 2009.

[14] L. Graftieaux, M. Michard, and N. Grosjean. Combining piv, pod and vortex identification algorithms for the study of unsteady turbulent swirling flows. *Measurement Science and Technology*, 2001.

[15] G. Haller. An objective definition of a vortex. *Journal of Fluid Mechanics*, pages 1–26, 2005.

[16] A. Hertzmann. Machine learning for computer graphics: A manifesto and tutorial. In *Pacific Conference on Computer Graphics and Applications*, pages 22–26. IEEE Computer Society, 2003.

[17] J. C. R. Hunt, A. Way, and P. Moin. Eddies, stream, and convergence zones in turbulent flows. Center for Turbulence Research Report CTR-S88, Center for turbulence research, Standford University, 1988.

[18] J. Jeong and F. Hussain. On the identification of a vortex. *Journal of Fluid Mechanics*, 285:69–94, 1 1995.

[19] M. Jiang, R. Machiraju, and D. Thompson. Geometric verification of swirling features in flow fields. In *Proceedings of the Conference on Visualization '02*, VIS '02, pages 307–314, 2002.

[20] M. Jiang, R. Machiraju, and D. Thompson. Detection and visualization of vortices. In *The Visualization Handbook*, pages 295–309. Academic Press, 2005.

[21] J. Kniss and G. Wang. Supervised manifold distance segmentation. *Visualization and Computer Graphics, IEEE Transactions on*, pages 1637–1649, Nov 2011.

[22] B. Kohler, R. Gasteiger, U. Preim, H. Theisel, M. Gutberlet, and B. Preim. Semi-automatic vortex extraction in 4d pc-mri cardiac blood flow data using line predicates. *IEEE Transactions on Visualization and Computer Graphics*, pages 2773–2782, 2013.

[23] K.-L. Ma. Machine learning to boost the next generation of visualization technology. *Computer Graphics and Applications, IEEE*, pages 6–9, Sept 2007.

[24] M. Otto and H. Theisel. Vortex analysis in uncertain vector fields. *Computer Graphics Forum*, 31(3pt2):1035–1044, 2012.

[25] H.-G. Pagendarm, B. Henne, and M. Rutten. Detecting vortical phenomena in vector data by medium-scale correlation. In *Visualization '99. Proceedings*, pages 409–552, Oct 1999.

[26] R. Peikert and M. Roth. The "parallel vectors" operator - a vector field visualization primitive. In *IEEE Visualization*, pages 263–270, 1999.

[27] S. K. Robinson. Coherent motions in the turbulent boundary layer. *Annual Review of Fluid Mechanics*, 23(1):601–639, 1991.

[28] M. Roth. *Automatic Extraction of Vortex Core Lines and Other Line-Type Features for Scientific Visualization*. PhD Dissertation No. 13673, ETH Zurich, 2000. published by Hartung-Gorre Verlag, Konstanz, ISBN 3-89649-582-8.

[29] M. Roth and R. Peikert. A Higher-Order Method for Finding Vortex Core Lines. In *Proceedings IEEE Visualization 1998*, pages 143–150, Los Alamitos, CA, USA, 1998. IEEE Computer Society Press.

[30] 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*, pages 419–422, 1998.

[31] T. Schafhitzel, J. E. Vollrath, J. P. Gois, D. Weiskopf, A. Castelo, and T. Ertl. Topology-preserving lambda$_2$-based vortex core line detection for flow visualization. *Comput. Graph. Forum*, 2008.

[32] S. Stegmaier, U. Rist, and T. Ertl. Opening the can of worms: An exploration tool for vortical flows. In *IEEE Visualization*, page 59. IEEE Computer Society, 2005.

[33] D. S. Thompson, J. S. Nair, S. S. D. Venkata, R. K. Machiraju, M. Jiang, and G. Craciun. Physics-based feature mining for large data exploration. *Computing in Science and Engg.*, 4(4):22–30, 2002.

[34] X. Tricoche, C. Garth, G. L. Kindlmann, E. Deines, G. Scheuermann, M. Rtten, and C. D. Hansen. Visualization of intricate flow structures for vortex breakdown analysis. In *IEEE Visualization*, pages 187–194. IEEE Computer Society, 2004.

[35] T. Weinkauf, J. Sahner, H. Theisel, and H.-C. Hege. Cores of swirling particle motion in unsteady flows. *Visualization and Computer Graphics, IEEE Transactions on*, pages 1759–1766, Nov 2007.

[36] J. Z. Wu, A. K. Xiong, and Y. T. Yang. Axial stretching and vortex definition. *Physics of Fluids 17*, 2005.

[37] L. Zhang, Q. Deng, R. Machiraju, A. Rangarajan, D. Thompson, D. K. Walters, and H.-W. Shen. Boosting techniques for physics-based vortex detection. *Computer Graphics Forum*, 33(1):282–293, 2014.